



*Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Malaya*

TOOLS FOR JAWI CHARACTER RECOGNITION

Perpustakaan SKTM

Oleh:

**NURAINIL MOHD MUSTAPA
WEK 000 278**

**PENASIHAT: EN ZAIDI RAZAK
MODURATOR : EN YAMANI IDNA IDRIS**

TOOL FOR JAWI CHARACTER RECOGNITION

**NURAINIL MOHD MUSTAPA
WEK 000 278**

**PENASIHAT : EN ZAIDI RAZAK
MODURATOR : EN YAMANI IDNA IDRIS**

ABSTRAK

Jika kita lihat secara rambang, pengecaman imej digital adalah kajian yang telah banyak diterokai oleh ahli-ahli profesional dan para pengkaji dalam bidang pekomputeran. Kajian ini adalah bertujuan untuk memberi kemudahan kepada pemahaman bahasa dan sebagainya. Kepelbagaian bahasa dan tulisan yang wujud pada tamadun manusia ini telah menarik ramai pengkaji untuk mengkaji dengan lebih mendalam bagaimana komputer mampu dan boleh mengenali dan memahami tulisan tanpa perlu menggunakan tenaga manusia. Kajian-kajian ini adalah perlu untuk tujuan pengecaman tulisan amnya dan keseragaman proses penterjemahan khususnya. Hari demi hari teknologi yang semakin canggih dan berkembang maju telah digunakan bagi melengkapkan kajian yang sedia ada bagi mencapai matlamatnya.

Oleh yang demikian, Tuhan telah membuka pintu hati ini untuk mengkaji bagaimana proses ini berlaku dalam dunia digital. Tulisan bangsa Melayu telah dijadikan bahan kajian untuk mengkaji bagaimana imej bagi tulisan Jawi dapat dipecahkan kepada aksara-aksara tunggal. Kajian ini amat penting memandangkan proses penterjemahan yang dilakukan sekarang terhadap manuskrip-manuskrip, kitab-kitab ilmu, dan bahan-bahan yang berkaitan amat sukar memandangkan tulisan Jawi zaman dahulu tidak seragam dengan tulisan Jawi moden yang digunapakai kini. Selain daripada itu, kekurangan tenaga pakar serta keputusan penterjemahan yang tidak selaras dan masa yang terlalu lama diambil bagi menterjemahkan sesuatu dokumen menjadi sebab utama kepada wujudnya kajian ini.

PENGHARGAAN

Alhamdulillah, saya menadah seribu kesyukuran kepada Yang Maha Kuasa kerana dengan limpah kurnia, petunjuk, dan izin-Nya dapat saya menyiapkan Projek Ilmiah Tahap Akhir (WXES 3182) yang bertajuk '*Tool for Jawi Character Recognition*' yang berasaskan kepada Projek Ilmiah Tahap Awal. Segala kesukaran yang mendalam yang dihadapi sepanjang menyiapkan projek ini tidak dapat diatasi tanpa adanya usaha bersungguh-sungguh, doa, serta bimbingan pelbagai pihak. Kesabaran yang tinggi, kemahiran berfikir, dan pengorbanan amatlah diperlukan untuk menjayakan projek ini.

Dengan ini, saya ingin memberi setinggi penghargaan kepada yang terutama iaitu penasihat saya yang saya hormati, **Encik Zaidi Razak** dan moduratur saya, **Encik Yamani Idna Idris** yang telah memberi banyak tunjuk ajar dan bimbingan kepada saya dalam menjayakan projek ini. Tidak lupa kepada rakan-rakan seperjuangan yang turut menghulur tangan dalam penghasilan projek ini iaitu **Nur Hasinah Hassan, Nornita Adila Mustafa, Suzylarina Samsudin, Hazrin Ahmad, dan Razman Abdul Ghani** di atas komitmen, kerjasama, dan sumbangan pengetahuan yang diberikan. Tidak lupa kepada ibu bapa saya serta mereka yang terlibat secara langsung atau tidak langsung dalam menjayakan projek ini.

Akhir kata, di harap projek ini dapat memberi manfaat kepada semua. Sekian, terima kasih.

KANDUNGAN

ABSTRAK.....	i
PENGHARGAAN.....	ii
KANDUNGAN.....	iii
SENARAI RAJAH.....	vii
SENARAI JADUAL.....	xi

BAB 1 – PENGENALAN

1.1	PENGENALAN.....	1
1.2	OBJEKTIF.....	3
1.3	SKOP OBJEKTIF.....	4
1.4	JADUAL PERANCANGAN PROJEK (CARTA GANTT).....	5

BAB 2 – METODOLOGI

2.1	PENGENALAN KEPADA JAWI.....	6
2.2	SEJARAH.....	7
2.3	KAJIAN SEDIA ADA.....	8
2.3.1	KAJIAN 1	8
2.3.2	KAJIAN 2.....	11

BAB 3 – ANALISA SISTEM

3.1	ANALISA SISTEM.....	13
3.2	KEPERLUAN FUNGSIAN.....	15

3.3	KEPERLUAN BUKAN FUNGSIAN.....	16
3.4	KEPERLUAN PERKAKASAN DAN PERISIAN.....	17
3.4.1	KEPERLUAN PERKAKASAN.....	17
3.4.2	KERPERLUAN PERISIAN.....	17

BAB 4 – REKABENTUK SISTEM

4.1	REKABENTUK SISTEM.....	18
4.1.1	PENDIGITALAN IMEJ.....	19
4.1.2	PRA-PEMROSESAN.....	19
4.1.2.1	PENGURANGAN GANGGUAN.....	19
4.1.2.2	PENINGKATAN KUALITI IMEJ.....	20
4.1.3	PENSEGMENTAN.....	22
4.1.4	PENGECAMAN.....	22
4.1.4.1	PANGKALAN PENGETAHUAN.....	24
4.1.5	KEPUTUSAN.....	25
4.2	REKABENTUK ANTARAMUKA.....	26

BAB 5 – ALGORITMA PENGECAMAN

5.1	ALGORITMA PENGECAMAN.....	30
5.2	PEMROSESAN IMEJ.....	31
5.2.1	Membuka fail dan memaparkan imej aksara jawi.....	31
5.2.2	Pengurangan gangguan menggunakan penapis <i>median</i>	31
5.2.3	Proses threshold.....	31

5.2.4	Proses thinning.....	31
5.3	PROSES PENSEGMENAN.....	31
5.4	PENGECAMAN IMEJ.....	34
5.4.1	Pembacaan kod rantai.....	34
5.4.2	Run length coding.....	45
5.4.3	Pengecaman bit.....	46

BAB 6 – PENGUJIAN SISTEM

6.1	PENGUJIAN SISTEM.....	48
6.2	PEMROSESAN IMEJ.....	48
6.2.1	Pendigitalan Imej.....	48
6.2.2	Pengurangan Gangguan.....	49
6.2.3	Peningkatan Kualiti Imej.....	49
6.2.4	Threshold.....	50
6.2.5	Thinning.....	50
6.3	PROSES PENSEGMENAN.....	51
6.4	PROSES PENGECAMAN.....	53
6.4.1	Pembacaan kod rantai.....	53
6.4.2	Pembacaan run-length coding.....	57
6.4.3	Pengecaman bit.....	57

BAB 7 – PERBINCANGAN

7.1	PERBINCANGAN.....	59
-----	-------------------	----

RUJUKAN.....	62
--------------	----

LAMPIRAN

Rajah 1: Huruf Ha dan Dai yang hampir sama.....	10
Rajah 2: Bentuk huruf 'Ain' (ع) yang berbeza berdasarkan.....	13
Rajah 3: Huruf 'Wau' dan 'Ra' yang mempunyai bentuk yang hampir serupa.....	13
Rajah 4: Huruf Jawi yang menggunakan simbol yang sama tetapi berbeza erti.....	14
Rajah 5: Aliran proses bagi pemprindaan (m) dalam naskah Jawi.....	18
Rajah 7: Jenis sajak dan selaras pada 'Mawana' dalam.....	21
Rajah 8: Taburan Jawi yang melalaui proses pengiraan.....	23
Rajah 9: Ciri-ciri Jawi yang berkaitan kod maklumat.....	23
Rajah 10: Aliran perubahan dalam sistem maklumat Jawi.....	24
Rajah 11: Huruf 'ba' yang akan dipaparkan dalam.....	28
Rajah 12: Huruf 'ba' yang akan dipaparkan dalam.....	29
Rajah 13: Huruf 'ba' yang akan dipaparkan dalam.....	30
Rajah 14: Huruf 'ba' yang akan dipaparkan dalam.....	31
Rajah 15: Huruf 'ba' yang akan dipaparkan dalam.....	32
Rajah 16: Huruf 'ba' yang akan dipaparkan dalam.....	32
Rajah 17: Huruf 'ba' yang akan dipaparkan dalam.....	33
Rajah 18: Huruf 'ba' yang akan dipaparkan dalam.....	33
Rajah 19: Huruf 'ba' yang akan dipaparkan dalam.....	34
Rajah 20: Huruf 'ba' yang akan dipaparkan dalam.....	34

SENARAI RAJAH

Rajah 1 : Aksara-aksara Jawi.....	6
Rajah 2 : Huruf Ra dan Dal yang hampir sama.....	10
Rajah 3 : Bentuk huruf ‘Ain (ع) yang berlainan kedudukan.....	13
Rajah 4 : Huruf Wau dan Ra yang mempunyai bentuk yang hampir serupa.....	13
Rajah 5 : Huruf Jawi yang mempunyai bentuk yang sama tetapi berbeza bilangan titik.....	14
Rajah 6 : Aliran proses bagi pemprosesan imej tulisan tangan Jawi.....	18
Rajah 7 : Imej sebelum dan selepas proses thinning dilakukan.....	21
Rajah 8 : Tulisan Jawi yang melalui proses segmentasi.....	22
Rajah 9 : Gambarajah lapan arah kod rantai.....	23
Rajah 10 : Aksara tambahan dicam menggunakan kaedah pengecaman bit.....	24
Rajah 11: Imej huruf ‘ba’ yang akan dijadikan bahan ujian.....	48
Rajah 12: Imej adalah hasil dari fungsi penapis median.....	49
Rajah 13: Imej threshold.....	50
Rajah 14: Imej yang telah melalui fungsi thinning.....	51
Rajah15: Aksara asas.....	52
Rajah 16: Akasara Tambahan.....	52
Rajah 17: Gambarajah lapan arah kod rantai.....	53
Rajah 18: Input aksara asas dari modul pensegmenan.....	53
Rajah 19: Cara pembacaan sebenar baris dan lajur.....	54
Rajah 20: Cara pembacaan baris dan lajur bagi penentuan titik mula.....	54

Rajah 21: Aksara asas huruf ‘ba’ di mana piksel berwarna kelabu mewakili nilai piksel 0 (imej) manakala piksel berwarna putih mewakili nilai 1. Anak panah merupakan bacaan kod rantai.....55

Rajah 22: Nilai piksel-piksel yang mungkin bagi piksel 8-kejiranan.....55

Rajah 23: Piksel yang berwarna hitam adalah piksel yang tidak mungkin mengandungi imej. Arah anak panah menunjukkan arah-arrah seterusnya yang mungkin mengikut bacaan arah jam merujuk kepada Jadual 2.....57

Rajah 24: Warna kelabu melambangkan imej aksara tambahan yang mewakili dua titik. Nilai 0 dan 1 di dalam piksel melambangkan nilai yang dipegang oleh setiap piksel di mana 0 mewakili warna hitam (imej) dan 1 mewakili warna putih....58

SENARAI JADUAL

Jadual 1: Titik-titik seterusnya yang mungkin bagi mendapatkan arah kod
rantai berdasarkan titik mula.....56

Jadual 2: Arah seterusnya yang mungkin berdasarkan kepada arah terakhir
Dibaca.....57

BAB 1

Pengenalan

University of Malaya

BAB 1

Pengenalan

1.1 PENGENALAN

Kajian terhadap penerimaan aksara dan objek-objek atau bentuk-bentuk tulisan tangan oleh sesebuah komputer atau character recognition telah lama dilakukan. Perbezaan bahasa dan tulisan hasil daripada negara, bangsa, agama, dan budaya yang berlainan telah menggalakkan ramai pengkaji, pengaturcara, dan juruanalisa sistem menyelami bagaimana komputer digunakan untuk mengenali dan mengecam bahasa-bahasa ini.

Pelbagai karya telah dihasilkan dengan menggunakan tulisan Jawi dalam bidang kesusasteraan di Malaysia. Sebelum munculnya Bahasa Malaysia sebagai bahasa rasmi di Malaysia, tulisan Jawi merupakan teras kepada komunikasi dan penulisan bagi penduduk negara ini. Bukan sahaja buku-buku ditulis dalam tulisan Jawi, malah penggunaannya begitu meluas meliputi semua bidang. Namun, kini tulisan Jawi semakin dikurangkan dalam era moden dan canggih.

Pelbagai bahan ilmiah, monumen-monumen, dan artifek-artifek lama ditulis dalam tulisan Jawi namun sehingga kini masih belum terungkai maksud yang tersirat disebaliknya. Penulisan Jawi pada zaman dahulu hanya menggunakan tulisan tangan berbanding zaman serba moden ini dimana banyak perisian telah dikeluarkan yang mana tulisan Jawi boleh ditaip menggunakan komputer. Ini menyebabkan bentuk tulisan Jawi yang pelbagai dan tidak seragam.

Dengan ini, dengan bantuan tenaga pensyarah maka tercetuslah idea untuk membuat kajian terhadap aksara-aksara Jawi ini dan untuk membina alatan yang lengkap untuk tujuan pengecaman tulisan tangan Jawi. Kerumitan pengecaman tulisan Jawi zaman dahulu akan dapat diatasi dalam kajian ini. Dalam kajian ini, setiap input imej tulisan Jawi yang telah diimbas dengan mesin pengimbas (scanner) akan diteliti dan akan

melalui proses-proses tertentu. Setiap tulisan Jawi yang telah diinput akan melalui proses pengurangan gangguan dan peningkatan kualiti imej sebelum melalui proses pensegmenan atau pemecahan input imej tersebut kepada aksara-aksara tunggal. Seterusnya proses pengecaman akan dilakukan di mana komputer akan mengenali dan mengecam input imej tadi untuk mengeluarkan output aksara tunggal.

Semoga dengan adanya kajian ini diharap dapat membantu generasi akan datang dalam proses penterjemahan tulisan tangan Jawi dahulu dan moden kepada bentuk yang lebih mudah difahami. Dengan segala kemudahan dan teknologi yang ada dibantu oleh alat-alat dan perisian sedia ada di pasaran, diharap kajian ini akan mencapai matlamat dan akan berjaya.

1.2 OBJEKTIF

Untuk menjayakan projek ini, beberapa objektif telah dikenalpasti:

- ✓ Mengecam imej tulisan tangan Jawi dengan memaparkan sebagai aksara Jawi tunggal.
- ✓ Mendigitalkan imej tulisan tangan Jawi dan seterusnya meningkatkan kualiti imej dan mengurangkan gangguan.
- ✓ Melakukan proses pensegmenan dan pepadanan untuk tujuan pengecaman aksara.
- ✓ Membangunkan pangkalan pengetahuan untuk menyimpan kod rantai (chain code) dan imej digital bagi aksara tertentu.
- ✓ Memudahkan proses pengecaman bagi tulisan tangan Jawi yang kabur dan kurang jelas.

1.3 SKOP PROJEK

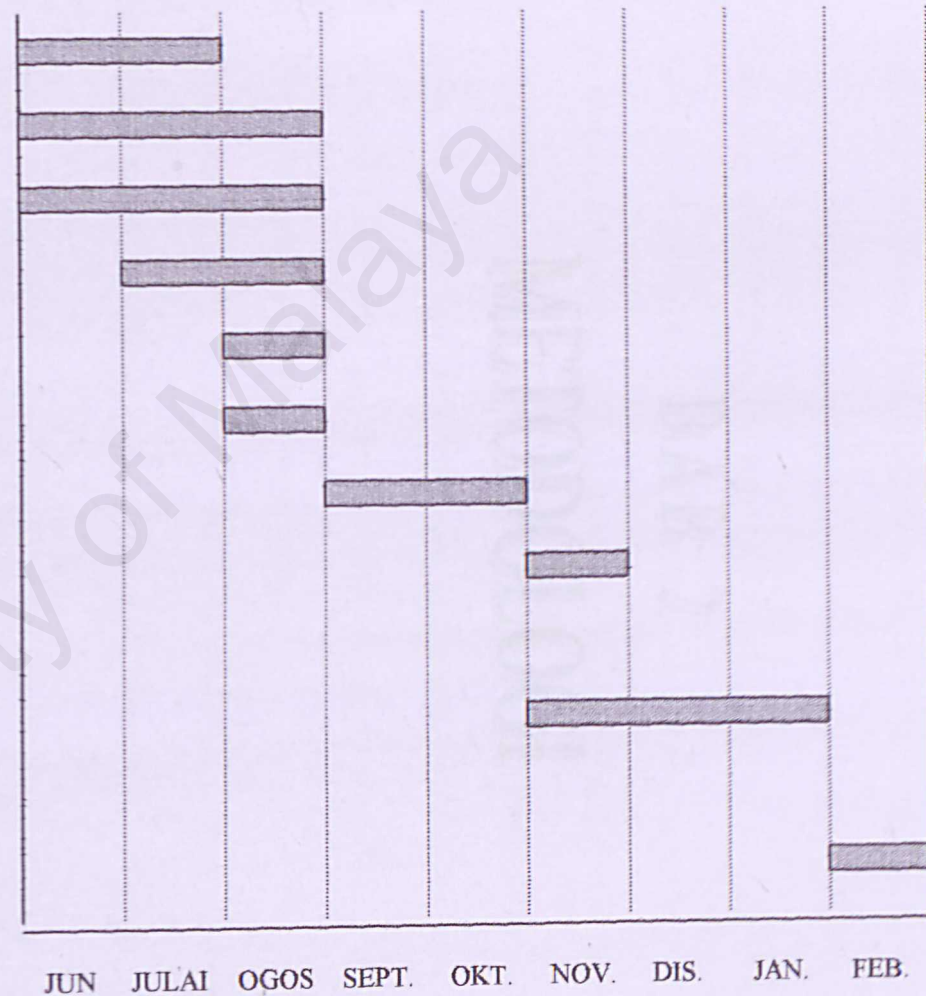
Skop bagi projek ini adalah seperti yang dinyatakan di bawah:

- ✓ Pengecaman dilakukan ke atas tulisan Jawi lama dan moden.
- ✓ Pengecaman hanya melibatkan satu patah perkataan Jawi yang diinput sahaja.
- ✓ Pemadanan bergantung kepada pangkalan pengetahuan yang mempunyai aksara asas dan aksara tambahan.
- ✓ Pengecaman dilakukan secara bukan talian-terus (off-line).
- ✓ Saiz imej yang akan diinput ditetapkan sebagai 512 x 512 piksel.

1.4 JADUAL PERANCANGAN PROJEK (CARTA GANNT)

AKTIVITI:

- Kajian dan pemahaman terhadap tajuk projek
- Mempelajari perisian Matlab yang akan digunakan.
- Memahami konsep pemprosesan imej secara teori.
- Kajian terhadap methodologi yang digunakan.
- Membuat spesifikasi proses rekabentuk
- Merekabentuk pangkalan pengetahuan
- Pemprosesan terhadap imej digital.
- Membangunkan pangkalan pengetahuan dengan memasukkan aksara yang dipilih.
- Membina aturcara program untuk proses pengecaman.
- Membuat ujian terhadap sistem yang telah dibina.



2.1 PENGENALAN KEPADA JAWI

Secara umumnya, tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri. Tulisan Jawi tidak mempunyai aksara besar dan kecil, cuma perincian bentuk kepala aksara-aksara sahaja ditulis secara berlainan atau bersambung atau berasingan.

Aksara Jawi adalah berbeza dari aksara tulisan lain, seperti tulisan Latin dan tulisan Cina, di mana tulisan Jawi ditulis dari kanan ke kiri.

Walaupun berbeza, tulisan Jawi mempunyai ciri-ciri yang sama dengan tulisan lain. Tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri.

Tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri. Tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri.

Tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri. Tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri.

BAB 2

METODOLOGI



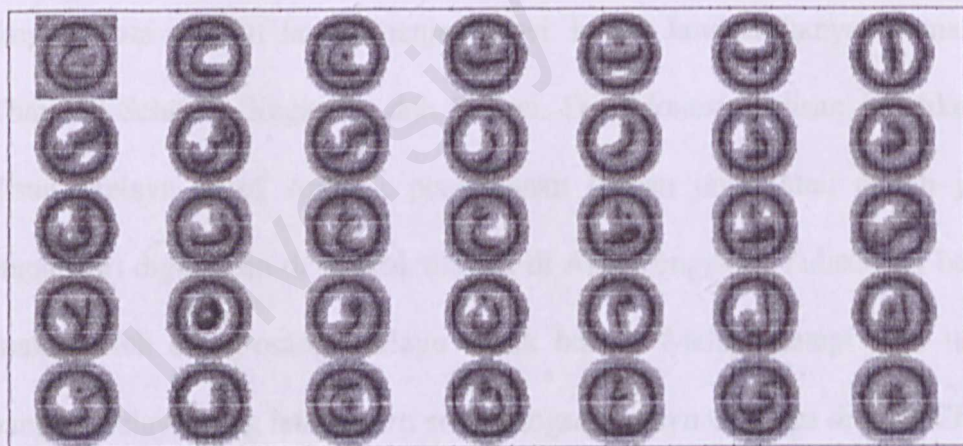
Gambar 2.1: Simbol-simbol Jawi

2.1 PENGENALAN KEPADA JAWI

Secara umumnya, tulisan Jawi adalah sejenis tulisan yang ditulis dari kanan ke kiri. Tulisan Jawi tidak mempunyai aksara besar atau kecil, cuma perubahan bentuk kepada aksara-aksara apabila ditulis secara tunggal atau bersambung atau berangkai.

Aksara Jawi adalah berbeza dari satu tulisan lain, seperti tulisan Latin dan tulisan Cina, di mana tulisan Jawi ditulis secara bersambung sama ada dalam bentuk cetakan atau tulisan tangan. Tulisan Jawi mengandungi 28 aksara utama. Walau bagaimanapun, kebanyakan daripada bentuknya berubah bergantung kepada kedudukan sesuatu aksara dalam sesuatu perkataan Jawi. Tambahan pula, sesetengah dari aksara-aksara Jawi ini mempunyai bentuk yang sama tetapi berbeza dari segi aksara tambahan iaitu satu, dua, atau tiga titik di atas atau di bawah aksara berkenaan.

Berikut merupakan 35 huruf Jawi yang digunakan untuk mengeja dan menulis kata-kata Melayu seperti yang ditunjukkan pada *Rajah 1* di bawah.



Rajah 1 : Aksara-aksara Jawi

2.2 SEJARAH

Mengikut sejarah, tulisan Jawi telah pun berkembang mulai tahun 1303 berdasarkan penemuan batu bersurat yang dijumpai di Terengganu Darul Iman. Dipercayai tulisan Jawi meresap ke dalam kebudayaan orang-orang Melayu bersama-sama dengan kedatangan agama Islam yang dipercayai dibawa oleh pedagang-pedangan Arab di Gugusan Kepulauan Melayu.

Sebelum kedatangan agama Islam, orang-orang melayu telah mempunyai kebudayaan dan tulisan mereka sendiri yang dipengaruhi oleh unsur-unsur Hindu. Antara contoh tulisan mereka yang terkenal pada masa itu ialah Renchong dan Jawi Kuno atau Kawi.

Tulisan Jawi ialah sejenis tulisan yang dipinjam oleh orang-orang melayu daripada tulisan Arab. Nama Jawi ini berasal daripada bahasa Arab "Jawah". Apabila ditukarkan kepada kata adjektif Jawah menjadi Jawi. Istilah Jawi ini hanya dikenali di Malaysia, Thailand Selatan, Singapura dan Brunei. Di Indonesia tulisan ini dikenali dengan "tulisan Melayu huruf Arab" . penggunaan tulisan jawi (atau istilah yang semakna dengannya) digunakan di banyak tempat di Asia Tenggara. Tulisan ini bukan sahaja digunakan oleh orang-orang Melayu untuk bahasa Melayu tetapi juga untuk suku-suku bangsa Melayu yang lain seperti suku bangsa Melayu Champa di Indo China, Patani di Thailand Selatan, Jawa, Banjar, Aceh Minangkabau, Bengkulu dan lain-lain.

2.3 KAJIAN SEDIA ADA

Untuk menjayakan projek ini, beberapa kajian telah dijalankan ke atas beberapa kajian terdahulu yang mempunyai persamaan dengan projek pengecaman tulisan Jawi. Antara kajian yang mempunyai persamaan dengan pengecaman aksara adalah bahasa Cina, bahasa Jepun, bahasa Arab, dan banyak lagi. Kajian terhadap pengecaman bahasa Arab tidak kurang banyaknya. Maka untuk mendapatkan metod yang sesuai bagi kajian ini, beberapa kajian terdahulu telah dikaji.

Oleh yang demikian, beberapa kajian telah dipilih untuk mengaplikasikan kaedah dan metod yang digunakan.

2.3.1 KAJIAN 1

Tajuk : Segmentation and Recognition of Arabic Characters by Structural Classification

Kajian oleh : B.M.F. Bushofa, M.Spann

METODOLOGI YANG DIGUNAKAN

Dalam kajian ini, proses pengecaman mempunyai beberapa langkah iaitu pepohon keputusan. Dalam kaedah ini, kesemua aksara pertama sekali dibahagikan kepada beberapa kumpulan bergantung kepada bilangan primitifnya. Kemudian, untuk setiap kumpulan jenis primitifnya (iaitu garisan, poligon terbuka, dan poligon tertutup) digunakan untuk membahagikannya kepada subkumpulan pula. Jika subkumpulan tersebut mempunyai aksara yang mempunyai jenis primitif yang sama, maka ia akan dipecahkan melalui arah garisan atau poligon terbuka. Bilangan tepi poligon digunakan

bagi membezakan antara aksara-aksara. Klasifikasi terakhir ialah dengan membawa maklumat tentang aksara tambahan yang disimpan di dalam pangkalan pengetahuan.

Langkah kedua yang digunakan ialah pemadanan jarak minimum. Dalam peringkat ini, jika aksara ditolak atau tidak terima oleh pepohon keputusan, kesemua ciri-ciri pada aksara tersebut akan disimpan dalam tatasusunan dan akan digunakan bagi mengira jarak City Block antara ciri-cirinya dengan ciri-ciri bagi setiap set bagi templat yang tertakrif.

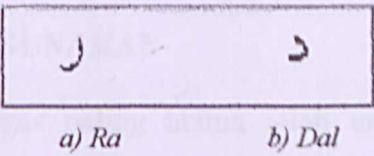
Untuk meminimumkan masa bagi proses pemadanan, bilangan primitif dibandingkan terlebih dahulu dengan templat. Jika perbezaannya adalah sama dengan sifar atau satu, maka jarak antara semua ciri akan dikira. Dan jika sebaliknya berlaku, perbezaannya adalah disetkan kepada nombor yang tinggi untuk memastikan pemadanan, dan templat seterusnya akan diambil kira. Setiap ciri pada aksara akan dipadankan dengan setiap ciri dalam set templat.

HASIL YANG DIPEROLEHI

Daripada kajian ini, pengecaman akan menggunakan aksara pra-pensegmenan. Jumlah bagi 20 ciri-ciri untuk setiap aksara primitif diasingkan daripada proses penskeletonan. Tiga dari ciri-ciri ini, iaitu jenis, arah, dan bilangan primitif, bersama aksara tambahan digunakan dalam pepohon keputusan. Jika aksara tidak dicam semasa pepohon keputusan, kesemua ciri ini akan dihantar ke proses Pemadanan Jarak Minimum untuk membandingkannya dengan set ciri yang piawai.

Walau bagaimanapun, terdapat kelemahan pada kajian ini di mana kebanyakan daripada kesalahan klasifikasi akan timbul antara aksara Ra (𐪗) dan Dal (𐪚) seperti

yang ditunjukkan pada *Rajah 2*. Walau pun ia dapat berbeza, skeletonnya adalah hampir sama. Apabila huruf Ra menjadi kecil, aksara Ra akan kelihatan seperti Dal. Manakala apabila huruf Dal menjadi besar, aksara Dal akan kelihatan seperti Ra.



Rajah 2 : Huruf Ra dan Dal yang hampir sama.

KESIMPULAN

Kertas persidangan ini telah memaparkan teknik baru untuk proses pensegmenan dan pengecaman bagi aksara Arab. Dalam teknik pensegmenan baru ini ia mengurangkan bilangan kelas bagi beberapa pemilihan bagi posisi pensegmenan. Algoritma aksara tambahan berkebolehan untuk mengenepikan kesemua aksara tambahan, di mana ia menyumbang kepada pengurangan bilangan klasifikasi aksara. Peranan paling penting dalam kajian ini ialah penskeletonan dan ciri-cirinya diasingkan dengan mengambil ukuran secara terus bagi aksara-aksara Jawi. Tambahan pula, ciri-ciri mengandungi segmen garisan dan poligon.

2.3.2 KAJIAN 2

Tajuk : Off-line Arabic Character Recognition – A Review

Kajian oleh : M. S. Khorsheed

METODOLOGI YANG DIGUNAKAN

Dalam kajian ini, tugas paling utama ialah untuk mengklasifikasikan objek kepada salah satu daripada beberapa kategori yang dinyatakan dalam kertas kajian ini. Terdapat beberapa teknik klasifikasi yang diaplikasikan dalam kertas kajian ini.

Antara metodologi yang digunakan dalam proses pengecaman adalah seperti Kajian 1 iaitu Pengklasifikasi Jarak Minimum dan Pengklasifikasi Pepohon Keputusan. Selain daripada itu, metodologi yang digunakan adalah Pengklasifikasi Statistik. Pengklasifikasi ini mengaggarkan beberapa kelas yang berbeza dan ciri-ciri vector yang mempunyai kemungkinan sambungan. Salah satu pendekatan yang digunakan ialah menggunakan Pengklasifikasi Bayes. Dalam Pengklasifikasi Bayes, ia meminimumkan jumlah purata kehilangan dalam mengecam corak yang tidak dapat dikenalpasti kepada satu kelas yang berkemungkinan.

Pengklasifikasi Al-Badr dan Haralick mengaplikasikan 3 langkah proses pengecaman: Pertama, sistem akan mencari bentuk primitif pada teks imej; kemudian ia akan mengesan primitif bagi perkataan. Pilihan akan menjadi salah satu daripada kemungkinan posteriori. Akhir sekali, kemungkinan bagi pepadanan akan dikira antara simbol model dan imej perkataan.

KESIMPULAN

Kertas kajian ini telah mengulas kepada kajian terdahulu terhadap pengecaman teks Arab. Dalam kajian ini, model am untuk pengecaman teks Arab telah diketengahkan. Model ini telah dibahagikan kepada lima tahap: perolehan imej, pra-pemprosesan, pensegmenan, pengasingan ciri, dan pengklasifikasi dan pengecaman.

3.1 ANALISA SISTEM

Salah satunya, tujuan Jawi merupakan tulisan yang unik dan agak rumit. Tujuan Jawi merupakan tulisan yang berkembang di masa akan-kelanya akan berkembang untuk memudahkan satu perkataan. Dengan ini, ia memudahkan proses pengiraan. Tambahan pula, ia ditulis dari kanan ke kiri.

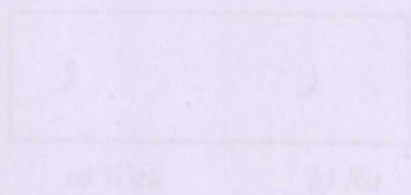
BAB 3

ANALISA SISTEM



Gambar 3.1: Contoh Jawi yang digunakan dalam proses pengiraan.

Tujuan Jawi Jawi lain seperti Jawi yang ditulis dengan penulisan yang lain. Maka inilah tujuan yang akan dipelajari dan bentuk-bentuk yang akan dipelajari. Bentuk Jawi yang hampir serupa antara satu sama lain akan dipelajari dan proses pengiraan. Seperti yang ditunjukkan dalam Gambar 3.1, huruf Wa ($و$) dan Wau ($و$) ini adalah bentuk yang hampir serupa yang mungkin akan memberi kegunaan yang sama untuk proses pengiraan.

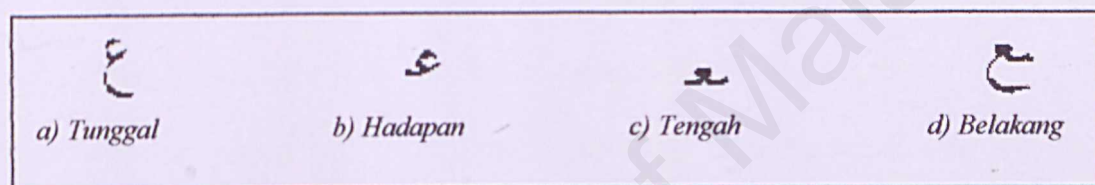


Gambar 3.2: Contoh Jawi yang digunakan dalam proses pengiraan.

3.1 ANALISA SISTEM

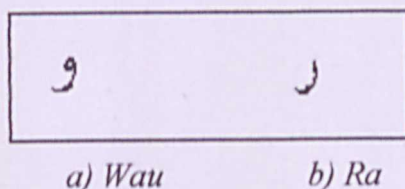
Secara amnya, tulisan Jawi merupakan tulisan yang unik dan agak rumit. Tulisan Jawi merupakan tulisan yang bersambung di mana aksara-aksaranya akan bersambung untuk membentuk satu perkataan. Dengan ini, ia merumitkan proses pengecaman. Tambahan pula, ia ditulis dari kanan ke kiri.

Dalam tulisan Jawi, sesuatu aksara mungkin akan mempunyai empat bentuk kedudukan iaitu tunggal, hadapan, tengah, dan belakang. Contohnya, seperti dalam *Rajah 3* di bawah, yang menggambarkan huruf 'Ain (ع) yang mempunyai aksara tunggal, hadapan, tengah, dan belakang.



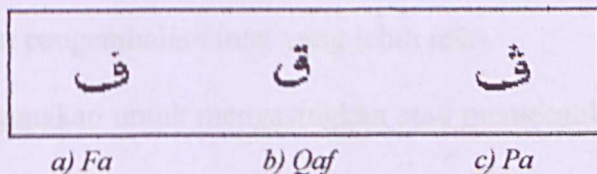
Rajah 3 : Bentuk huruf 'Ain (ع) yang berlainan kedudukan.

Tulisan tangan Jawi kebiasaannya berbeza antara seorang penulis dengan penulis yang lain. Maka masalah tulisan yang tidak sempurna penulisannya dari bentuk asalnya adalah mengelirukan. Bentuk aksara Jawi yang hampir serupa antara satu sama lain akan menyukarkan proses pengecaman. Seperti yang ditunjukkan dalam *Rajah 4*, huruf Ra (ر) dan Wau (و) mempunyai bentuk yang hampir serupa yang mungkin akan memberi keputusan yang kurang tepat semasa proses pengecaman.



Rajah 4 : Huruf Wau dan Ra yang mempunyai bentuk yang hampir serupa.

Ada juga terdapat huruf-huruf dalam tulisan Jawi yang mempunyai bentuk yang sama dengan huruf Jawi yang lain, tetapi huruf dan sebutannya berbeza. Ia berbeza dari sudut aksara tambahan yang wujud pada huruf Jawi tersebut, seperti bilangan titik dan hamzah dan sama ada ia berada di bahagian atas atau bawah sesuatu aksara Jawi. Contohnya huruf Fa (ف), Qaf (ق), dan Pa (ڤ) yang mempunyai bentuk yang sama tetapi bilangan titik yang berbeza seperti dalam *Rajah 5*.



Rajah 5 : Huruf Jawi yang mempunyai bentuk yang sama tetapi berbeza bilangan titik.

3.2 KEPERLUAN FUNGSIAN

Keperluan fungsian merupakan fungsi-fungsi atau modul yang terlibat dalam kajian ini. Keperluan fungsian bagi kajian ini adalah seperti yang dinyatakan di bawah:

- Input imej bagi kajian ini adalah dalam imej digital iaitu dalam bentuk bitmap seperti GIF, TIF, JPEG, HDF, DICOM, dan PNG.
- Proses pengurangan gangguan dan peningkatan kualiti imej meliputi proses menapis gangguan dan pengembalian imej yang lebih jelas.
- Proses segmentasi digunakan untuk mengasingkan atau memecahkan imej tulisan tangan Jawi kepada aksara-aksara tunggal.
- Pengecaman adalah proses terakhir di mana imej tulisan Jawi yang telah diasingkan akan dicam dan dipadankan dengan huruf Jawi sebenar.
- Output bagi kajian ini adalah imej huruf-huruf Jawi mewakili imej asal yang diinputkan.

3.3 KEPERLUAN BUKAN FUNGSIAN

Keperluan bukan fungsian adalah kekangan yang dihadapi dalam menjalankan kajian terhadap imej tulisan Jawi. Keperluan bukan fungsian disenaraikan seperti di bawah:

- Saiz imej yang diinput mestilah ditetapkan sebagai 512 x 512 piksel untuk mendapatkan imej yang lebih terang dan jelas.
- Format fail bagi menyimpan imej digital adalah dalam bentuk bitmap.
- Imej yang diinput mestilah berada dalam bentuk tahap kelabu (grayscale). Ini adalah untuk memudahkan penggunaannya dalam proses-proses dan teknik-teknik yang digunakan dalam kajian ini.

3.4 KEPERLUAN PERKAKASAN DAN PERISIAN

3.4.1 KEPERLUAN PERKAKASAN

Keperluan perkakasan yang diperlukan dalam kajian ini adalah seperti berikut:

- 255 Mhz Pemproses Pentium II
- 128.0 MB RAM
- 512 KB Internal L2 Cache
- 5 GB Cakera Keras

3.4.2 KEPERLUAN PERISIAN

Keperluan perisian yang diperlukan dalam kajian ini adalah seperti berikut:

- Microsoft Windows 98 / 2000
- MATLAB 6.1 Image Processing Tools

BAB 4

REKABENTUK SISTEM



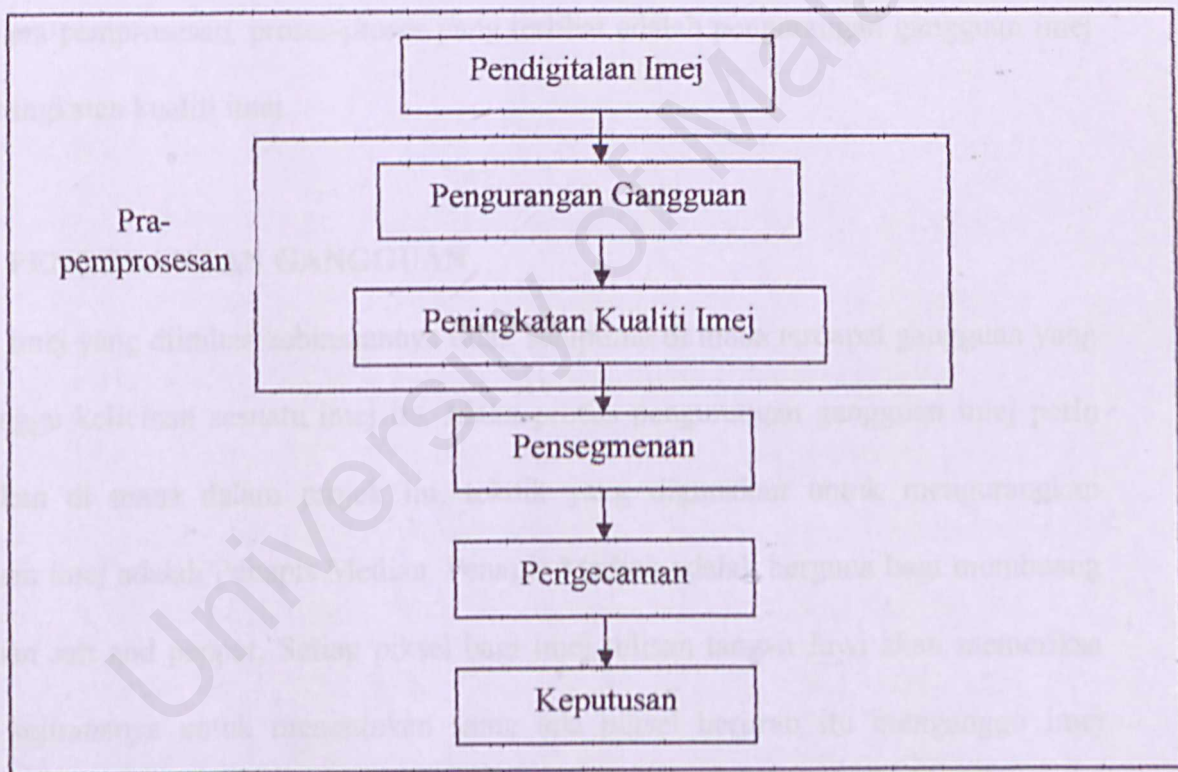
Diagram 4.1: Struktur Sistem (Rekabentuk Sistem) yang menunjukkan proses perancangan sistem.

Struktur sistem menunjukkan bagaimana data akan mengalir melalui proses-proses yang terlibat dalam sistem. Struktur sistem yang baik akan memudahkan pengguna untuk memahami dan menggunakan sistem.

4.1 REKABENTUK SISTEM

Setelah kajian dijalankan ke atas beberapa kertas kajian terdahulu, beberapa teknik telah dikenalpasti bagi melaksanakan alatan bagi projek ini. Teknik-teknik tersebut adalah pengurangan gangguan (*noise reduction*), peningkatan kualiti imej (*image enhancement*), pensegmenan (*segmentation*), dan pengecaman (*recognition*).

Teknik-teknik yang disebutkan di atas merupakan satu aliran proses bagi mencapai matlamat projek ini iaitu mengenali tulisan Jawi bagi menghasilkan aksara-aksara Jawi tunggal. Kita akan lihat proses-proses yang akan dilakukan ke atas tulisan tangab Jawi secara terperinci seperti pada *Rajah 6*.



Rajah 6 : Aliran proses bagi pemprosesan imej tulisan tangan Jawi.

Seksyen seterusnya akan menerangkan dengan lebih lanjut tentang proses-proses yang terlibat seperti *Rajah 3* di atas.

4.1.1 PENDIGITALAN IMEJ

Dalam proses ini, teks tulisan Jawi yang ingin dicamkan seperti dalam bentuk buku, artefak, risalah, dan sebagainya akan diimbas menggunakan mesin pengimbas (*scanner*). Imej tulisan tangan Jawi tersebut akan disimpan dalam bentuk bitmap seperti GIF, TIF, JPEG, HDF, DICOM, dan PNG. Imej ini kemudian akan disimpan di dalam fail yang telah ditetapkan dan akan menjalani proses seterusnya. Imej yang diimbas mestilah dalam bentuk penskalaan kelabu (*grayscale*).

4.1.2 PRA-PEMROSESAN

Selepas proses pendigitalan imej, proses pra-pemprosesan imej akan dijalankan. Dalam pra-pemprosesan, proses-proses yang terlibat adalah pengurangan gangguan imej dan peningkatan kualiti imej.

4.1.2.1 PENGURANGAN GANGGUAN

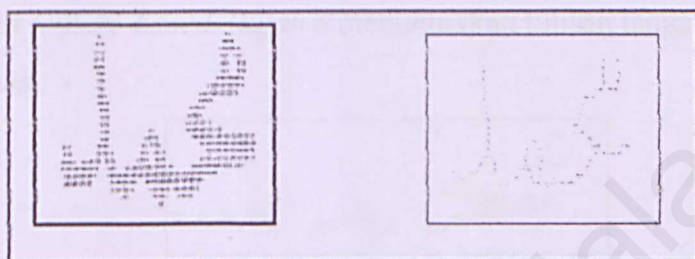
Imej yang diimbas kebiasaannya tidak sempurna di mana terdapat gangguan yang mengganggu kelicinan sesuatu imej itu. Maka proses pengurangan gangguan imej perlu dijalankan di mana dalam projek ini, teknik yang digunakan untuk mengurangkan gangguan imej adalah Penapis Median. Penapis Median adalah berguna bagi membuang gangguan salt and pepper. Setiap piksel bagi imej tulisan tangan Jawi akan memeriksa piksel kejirannya untuk menentukan sama ada piksel berjiran itu mengganggu imej tersebut atau tidak. Penapis Median akan menggantikan nilai piksel tersebut dengan nilai median bagi kesemua nilai piksel pada imej tersebut.

4.1.2.2 PENINGKATAN KUALITI IMEJ

Selepas proses pengurangan gangguan, imej perlu menjalankan proses peningkatan kualiti imej. Imej akan melalui proses ini untuk mencantikkan imej yang kabur hasil dari imej yang diimbas. Selain daripada itu, imej yang terhasil daripada imbasan mungkin akan mengandungi pelbagai kekurangan. Antaranya ialah tulisan yang bercelaru iaitu terdapat getaran tangan pada tulisan, pertindihan tulisan, dan juga imej yang kurang jelas di mana tulisan tersebut mungkin hasil daripada buku-buku dan artifek-artifek lama.

Dalam proses peningkatan kualiti imej, terdapat beberapa proses yang akan dijalankan ke atas imej. Proses pertama ialah mengubah tahap kecerahan imej (*brightness adjustment*). Dalam proses ini, imej akan diubah tahap kecerahannya di mana imej mungkin tidak jelas atau gelap dan proses ini akan mengubah tahap kecerahan sesuatu imej supaya kelihatan lebih jelas. Kemudian, proses melicinkan tepian imej (*edge smoothing*) akan dijalankan. Dalam proses ini, imej mungkin mempunyai tepian yang bergerigi atau tidak rata. Maka proses melicinkan tepian imej akan dijalankan supaya imej akan lebih licin dan cantik bagi memudahkan proses seterusnya. Seterusnya, proses *threshold* akan dilakukan di mana imej berada dalam bentuk warna (RGB) dan imej tahap kelabu (*grayscale*) akan ditukarkan kepada imej penduaan (*binary*) atau imej hitam putih. Piksel yang mempunyai imej hitam akan mewakili nilai 0 manakala piksel yang mempunyai imej putih akan mewakili nilai 1. Nilai *threshold* dalam kajian ini telah ditetapkan iaitu 180. Ini adalah bertujuan untuk memudahkan proses pensegmenan yang akan dijalankan seterusnya.

Akhir sekali, proses *thinning* akan dilakukan. Proses ini perlu dilakukan bagi memudahkan proses segmentasi dan pengecaman. Dalam proses ini, teknik yang akan digunakan ialah teknik *erosion* dan *skeleton* yang akan menggunakan penstrukturan elemen. Proses *skeleton* akan mengurangkan imej asal kepada imej bentuk garisan. Proses *erosion* pula digunakan untuk mengecilkan sempadan imej. Rajah 7 di bawah menunjukkan tulisan tangan Jawi yang melalui proses thinning.



a) Sebelum

b) Selepas

Rajah 7 : Imej sebelum dan selepas proses thinning dilakukan

4.1.3 PENSEGMENAN

Proses seterusnya adalah proses pensegmenan. Proses pensegmenan bertujuan untuk memecahkan tulisan Jawi yang bersambung kepada tidak bersambung. Dalam proses ini, *edge detection* akan digunakan bagi mengesan sempadan bagi sesuatu imej. Dalam *edge detection*, teknik yang akan dilakukan *Sobel*. Kaedah *Sobel* digunakan bagi mengesan tepian imej tulisan Jawi yang bersambung dan tidak bersambung.

Kemudian, imej tulisan Jawi yang bersambung akan diasingkan menggunakan metod *structural feature-based*. Rajah 8 menunjukkan tulisan tangan Jawi yang melalui proses segmentasi.



Rajah 8 : Tulisan Jawi yang melalui proses segmentasi

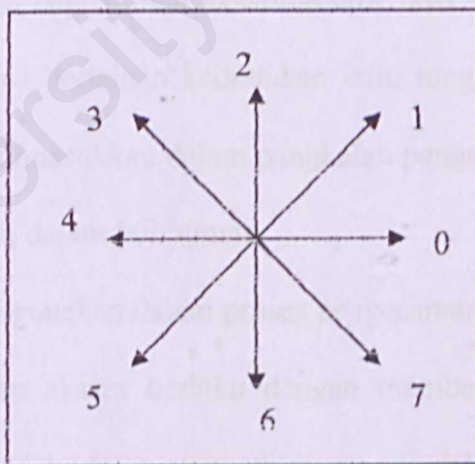
4.1.4 PENGECAMAN

Proses pengecaman merupakan proses akhir bagi kajian ini. Tulisan Jawi yang telah diasingkan kepada aksara-aksara dalam proses segmentasi akan melalui proses pengecaman. Proses pengecaman akan menentukan sama ada jenis huruf yang telah dipecahkan wujud atau tidak. Teknik yang digunakan dalam proses pengecaman adalah kod rantai atau *chain code*. Kod rantai adalah jujukan nombor-nombor yang mewakili bentuk asas aksara-aksara berdasarkan gambarajah lapan arah seperti yang ditunjukkan pada Rajah 9. Gambarajah lapan arah kod berantai digunakan kerana imej yang diwakilinya akan kelihatan lebih licin dan sempurna. Setiap bentuk aksara akan diwakili oleh nombor-nombor dan arah dalam gambarajah tersebut. Cara bacaannya adalah mengikut putaran jam. Setelah kod rantai diperolehi hasil daripada pembacaan

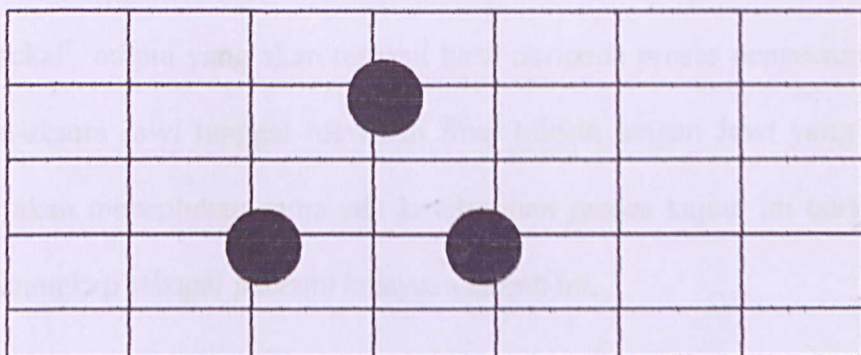
gambarajah arah bagi sesuatu aksara, proses pemadanan akan dilakukan untuk membandingkan kod rantaian yang disimpan dalam pangkalan pengetahuan dengan kod rantaian yang dibaca daripada aksara tersebut. Proses pemadanan ini dikenali sebagai *Dynamic Programming*.

Kaedah kedua yang digunakan adalah kaedah pengecaman bit. Kaedah pengecaman bit ini adalah untuk mengesan aksara tambahan. Aksara tambahan adalah terdiri daripada titik (sama ada satu, dua, atau tiga) dan hamzah (ء) sama ada di atas, tengah, atau di bawah aksara. Oleh kerana aksara tambahan ini bukan merupakan garisan bersambung atau bentuk, kaedah menentukan sama ada terdapat titik adalah dengan mengenalpasti setiap koordinat yang mengandungi imej atau titik seperti yang ditunjukkan dalam *Rajah 10*.

Namun begitu, bagi aksara tambahan hamzah, kaedah kod rantaian akan digunakan memandangkan hamzah mempunyai bentuk.



Rajah 9 : Gambarajah lapan arah kod rantaian



Rajah 10 : Aksara tambahan dicam menggunakan kaedah pengecaman bit.

4.1.4.1 PANGKALAN PENGETAHUAN

Pangkalan pengetahuan (*knowledge base*) amat diperlukan dalam proses pengecaman. Pangkalan pengetahuan akan dibina sebelum imej tulisan Jawi yang diinput menjalani proses-proses dalam kajian ini.

Pangkalan pengetahuan merupakan kamus bagi proses pengecaman yang akan menyimpan kod rantaian (*chain code*) bagi semua aksara-aksara Jawi yang mungkin sama ada aksara asas dari aksara Jawi berlainan kedudukan iaitu tunggal, belakang, tengah, dan hadapan. Data-data yang dimasukkan dalam pangkalan pengetahuan adalah kod rantaian dan bukannya bentuk imej dalam fail bitmap.

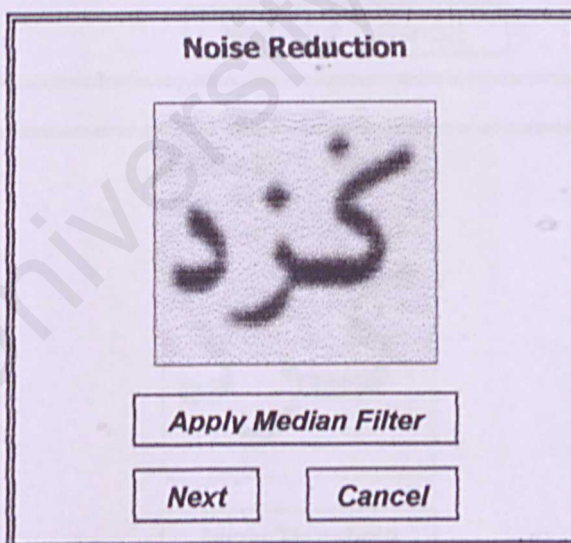
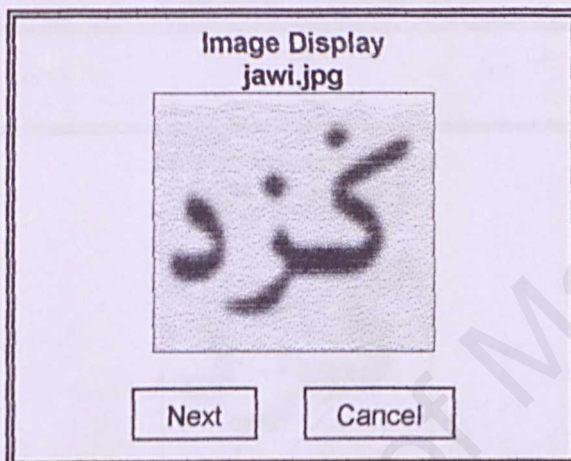
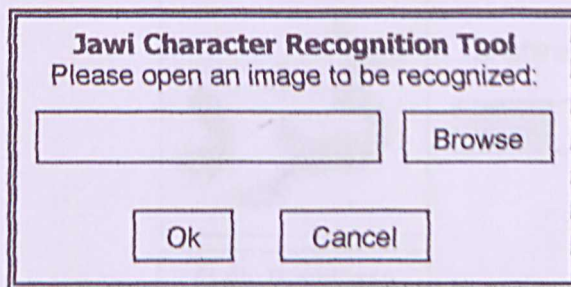
Pangkalan pengetahuan akan digunakan dalam proses pengecaman dalam proses pemadanan aksara. Proses pemadanan aksara berlaku dengan membandingkan kod rantaian pada aksara imej dengan kod rantaian yang disimpan di dalam pangkalan pengetahuan.

4.1.5 KEPUTUSAN

Akhir sekali, output yang akan terhasil hasil daripada proses pengecaman ialah paparan aksara-aksara Jawi tunggal mewakili imej tulisan tangan Jawi yang diinput.

Keputusan ini akan menentukan sama ada keseluruhan proses kajian ini berjaya atau tidak dan ia merangkap sebagai penentu kejayaan kajian ini.

4.2 REKABENTUK ANTARAMUKA



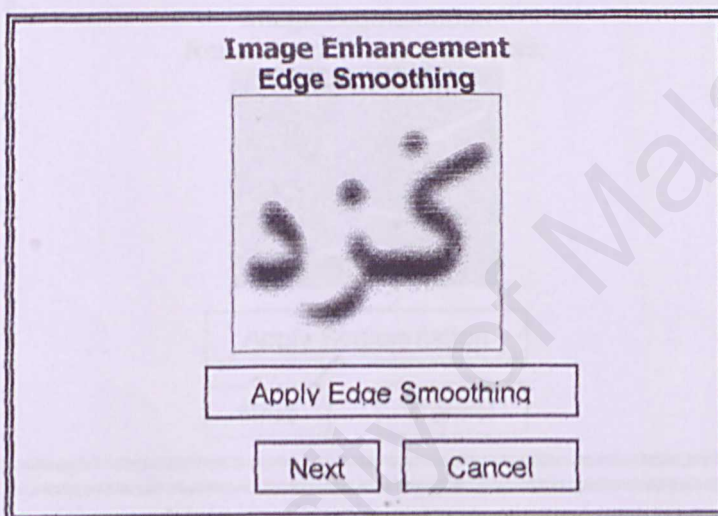
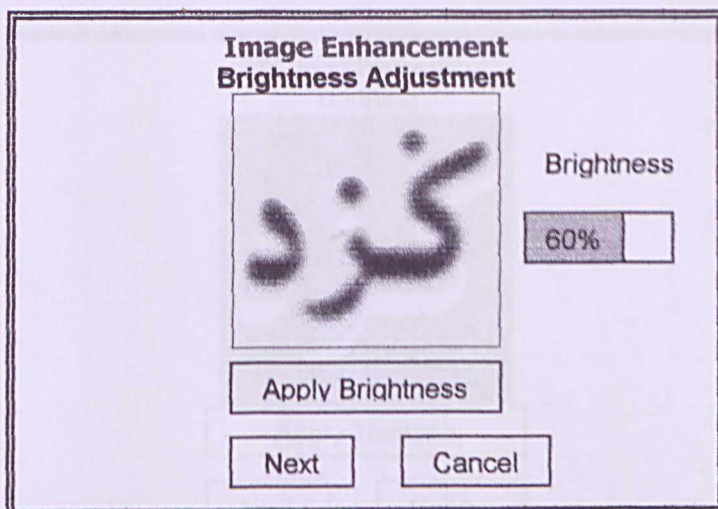



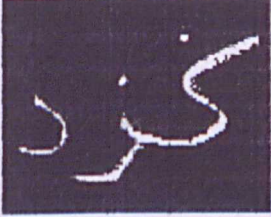
Image Enhancement
Thinning



Apply Thinning

Next Cancel


Image Segmentation
Result from Thinning process:



Apply Segmentation

Next Cancel

Image Recognition
Result from Segmentation process:



Apply Recognition

Next Cancel



ALGORITMA PENGENALAN
University of Malaya

BAB 5

ALGORITMA PENGECEMAN

5.1 ALGORITMA PENGECAMAN

Algoritma pengecaman imej terbahagi kepada modul-modul mengikut aliran proses seperti yang telah dinyatakan di dalam Rajah 6. Modul-modul tersebut adalah modul pemprosesan imej, proses psegmenan, dan modul pengecaman bagi satu huruf jawi.

Di dalam modul pemprosesan imej, imej huruf jawi disimpan di dalam direktori akan dibuka dan dipaparkan. Saiz bagi imej tersebut akan ditukar kepada saiz yang ditetapkan. Seterusnya modul ini melaksanakan pengurangan gangguan pada imej dengan menggunakan penapis *median*. Penapis median digunakan bagi mengurangkan gangguan *salt and papper*. Kemudian peningkatan kualiti imej (*brightness adjustment*) akan dilakukan bagi menghasilkan imej yang lebih berkualiti. Modul ini juga merangkumi proses *thinning* dan *threshold*.

Modul psegmenan dijalankan terhadap satu aksara jawi yang bertujuan untuk mengasingkan aksara asas dan aksara tambahan. Seperti yang telah sedia maklum, aksara asas merupakan struktur asas bagi satu aksara jawi manakala aksara tambahan menggambarkan titik yang menentukan identiti aksara berkenaan.

Seterusnya, modul pengecaman terdiri daripada penentuan titik mula, pembacaan kod rangkaian, dan pepadanan aksara dengan pangkalan pengetahuan bagi aksara. Titik mula perlu ditentukan bagi tujuan pembacaan kod rangkaian. Setelah kod rangkaian dibaca bagi aksara asas, ia perlu disimpan dan dibandingkan dengan pangkalan pengetahuan yang telah dibina.

5.2 PEMROSESAN IMEJ

5.2.1 Membuka fail dan memaparkan imej aksara jawi

```
[imagefile,imagepath]=uigetfile('*. *','Select your image');  
[I,map]=imread([imagepath,imagefile]);  
image(I);
```

5.2.2 Pengurangan gangguan imej menggunakan penapis *median*

```
B=rgb2gray(I);  
noise=medfilt2(B, [3 3]);
```

5.2.3 Peningkatan kualiti imej

```
beta = ++.5; brighten(beta);  
beta = .3; brighten(-beta);
```

5.2.4 Proses *threshold*

```
level=graythresh(noise);  
threshold = im2bw(noise,level);
```

5.2.5 Proses *thinning*

```
SE = strel('square',2);  
thinning = imdilate(threshold,SE)
```

5.3 PROSES PENSEGMENAN

```
[m,n] = size(thinning);  
bw=I;  
length_array=length(bw);  
array = ones(1,length_array);  
M = cell(length_array,1);  
aksara_asas = cell(length_array,1);  
aksara_tambahan=cell(length_array,1);
```

% ----- Masukkan imej (binary=0) ke dalam cell array M ----- %

for i=1:m

total=0;

for j=1:length_array

if bw(i,j)==1 %Tentukan bilangan j=1

total=total+1;

end

end

if total ==length_array

total=0;

elseif total~=length_array

for a=1:length_array

array(1,a)=bw(i,a);

end

M{i} = array;

total=0;

end

end

% ----- Proses Segmentation ----- %

temp=0;

temp1=0;

temp2=0;

temp3=0;

temp4=0;

temp5=0;

for i=1:m

see=isempty(M{i});

if see==1 %kalau M{i} tidak mempunyai nilai 0

temp=temp+1;

```

    temp1=temp;

elseif see==0 %kalau M{i} mempunyai nilai 0
    temp=temp+1;
    temp2=temp;
    break
end
end

temp3=temp2;
array2=ones(1,n);
aksara_asas{temp3-1} *array2;
for i=temp3:m
    see isempty(M{i});
    if see==0
        aksara_asas{i}=M{i};
    else
        aksara_asas{i}=array2;
        temp4=temp4+i;
        break
    end
end
temp5=temp4;
for i=temp5:m
    aksara_tambahan{i}=M{i};
end

y = cell2mat(aksara_asas);
figure,imshow(y),title('Aksara Asas');

z=cell2mat(aksara_tambahan);
figure,imshow(z);

```


5.4 PENGECAMAN IMEJ

5.4.1 Pembacaan kod rantaian

```
[k,l]=size(y);

% ----- Determine initial point for image ----- %
length_array=length(l);
for i=1:k
    for j=l:-1:1
        if I(i,j)==0
            start1=i;
            start2=j;
            disp(l)
            fprintf(1,'Titik mula bagi imej adalah pada piksel (%.1d , %.1d )
\n',start1,start2)
            break;
        end
    end
    if I(i,j)==0
        break;
    end
end
prev_code=8;
% ----- Determine the chain code for image ----- %

store_code=[];
increment=1;
while start1~=0 & start2~=0
    if prev_code==8
        if I(start1+1,start2-1)==0 %code=5
            start1=start1+1;
            start2=start2-1;
```

```
store_code{increment}=5;
prev_code=5;
elseif I(start1+1,start2)==0 %code=6
    start1=start1+1;
    start2=start2;
    store_code{increment}=6;
    prev_code=6;
elseif I(start1-1,start2)==0 %code=2
    start1=start1-1;
    start2=start2;
    store_code{increment}=2;
    prev_code=2;
elseif I(start1,start2+1)==0 %code=0
    start1=start1;
    start2=start2+1;
    store_code{increment}=0;
    prev_code=0;
elseif I(start1-1,start2+1)==0 %code=1
    start1=start1-1;
    start2=start2+1;
    store_code{increment}=1;
    prev_code=1;
elseif I(start1-1,start2-1)==0 %code=3
    start1=start1-1;
    start2=start2-1;
    store_code{increment}=3;
    prev_code=3;
elseif I(start1,start2-1)==0 %code=4
    start1=start1;
    start2=start2-1;
    store_code{increment}=4;
```

```

    prev_code=4;
elseif I(start1+1,start2+1)==0 %kod=7
    start1=start1+1;
    start2=start2+1;
    store_code{increment}=7;
    prev_code=7;
else
    start1=0;
    start2=0;
end
increment=increment+1;
else % ***** if prev_code is exist - after defining the previous code
    if prev_code==5 % ***** if prev_code==5
        if I(start1+1,start2+1)==0 %kod=7 - prev_code=5
            start1=start1+1;
            start2=start2+1;
            store_code{increment}=7;
            prev_code=7;
        elseif I(start1+1,start2)==0 %code=6 - prev_code=5
            start1=start1+1;
            start2=start2;
            store_code{increment}=6;
            prev_code=6;
        elseif I(start1+1,start2-1)==0 %code=5 - prev_code=5
            start1=start1+1;
            start2=start2-1;
            store_code{increment}=5;
            prev_code=5;
        elseif I(start1,start2-1)==0 %code=4 - prev_code=5
            start1=start1;
            start2=start2-1;

```



```
store_code{increment}=4;
prev_code=4;
elseif I(start1-1,start2-1)==0 %code=3 - prev_code=5
    start1=start1-1;
    start2=start2-1;
    store_code{increment}=3;
    prev_code=3;
elseif I(start1-1,start2)==0 %code=2 - prev_code=5
    start1=start1-1;
    start2=start2;
    store_code{increment}=2;
    prev_code=2;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==6 % ***** if prev_code==6

if I(start1+1,start2+1)==0 %kod=7 - prev_code=6
    start1=start1+1;
    start2=start2+1;
    store_code{increment}=7;
    prev_code=7;
elseif I(start1+1,start2)==0 %code=6 - prev_code=6
    start1=start1+1;
    start2=start2;
    store_code{increment}=6;
    prev_code=6;
elseif I(start1+1,start2-1)==0 %code=5 - prev_code=6
    start1=start1+1;
```

```

    start2=start2-1;
    store_code{increment}=5;
    prev_code=5;
elseif I(start1,start2-1)==0 %code=4 - prev_code=6
    start1=start1;
    start2=start2-1;
    store_code{increment}=4;
    prev_code=4;
elseif I(start1-1,start2-1)==0 %code=3 - prev_code=6
    start1=start1-1;
    start2=start2-1;
    store_code{increment}=3;
    prev_code=3;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==2 % ***** if prev_code==2
    if I(start1-1,start2-1)==0 %code=3 - prev_code=2
        start1=start1-1;
        start2=start2-1;
        store_code{increment}=3;
        prev_code=3;
    elseif I(start1-1,start2)==0 %code=2 - prev_code=2
        start1=start1-1;
        start2=start2;
        store_code{increment}=2;
        prev_code=2;
    elseif I(start1-1,start2+1)==0 %code=1 - prev_code=2
        start1=start1-1;

```

```

    start2=start2+1;
    store_code{increment}=1;
    prev_code=1;
elseif I(start1,start2+1)==0 %code=0 - prev_code=2
    start1=start1;
    start2=start2+1;
    store_code{increment}=0;
    prev_code=0;
elseif I(start1+1,start2+1)==0 %kod=7 - prev_code=2
    start1=start1+1;
    start2=start2+1;
    store_code{increment}=7;
    prev_code=7;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==0 % ***** if prev_code==0
    if I(start1-1,start2+1)==0 %code=1 - prev_code=0
        start1=start1-1;
        start2=start2+1;
        store_code{increment}=1;
        prev_code=1;
    elseif I(start1,start2+1)==0 %code=0 - prev_code=0
        start1=start1;
        start2=start2+1;
        store_code{increment}=0;
        prev_code=0;
    elseif I(start1+1,start2+1)==0 %kod=7 - prev_code=0
        start1=start1+1;

```



```

    start2=start2+1;
    store_code{increment}=7;
    prev_code=7;
elseif I(start1+1,start2)==0 %code=6 - prev_code=0
    start1=start1+1;
    start2=start2;
    store_code{increment}=6;
    prev_code=6;
elseif I(start1+1,start2-1)==0 %code=5 - prev_code=0
    start1=start1+1;
    start2=start2-1;
    store_code{increment}=5;
    prev_code=5;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==1 % ***** if prev_code==1
    if I(start1-1,start2-1)==0 %code=3 - prev_code=1
        start1=start1-1;
        start2=start2-1;
        store_code{increment}=3;
        prev_code=3;
    elseif I(start1-1,start2)==0 %code=2 - prev_code=1
        start1=start1-1;
        start2=start2;
        store_code{increment}=2;
        prev_code=2;
    elseif I(start1-1,start2+1)==0 %code=1 - prev_code=1
        start1=start1-1;

```

```

    start2=start2+1;
    store_code{increment}=1;
    prev_code=1;
elseif I(start1,start2+1)==0 %code=0 - prev_code=1
    start1=start1;
    start2=start2+1;
    store_code{increment}=0;
    prev_code=0;
elseif I(start1+1,start2+1)==0 %kod=7 - prev_code=1
    start1=start1+1;
    start2=start2+1;
    store_code{increment}=7;
    prev_code=7;
elseif I(start1+1,start2)==0 %code=6 - prev_code=1
    start1=start1+1;
    start2=start2;
    store_code{increment}=6;
    prev_code=6;
else
    start1=0;
    start2=0;
end
increment=increment+1
elseif prev_code==3 % ***** if prev_code==3
    if I(start1+1,start2-1)==0 %code=5 - prev_code=3
        start1=start1+1;
        start2=start2-1;
        store_code{increment}=5;
        prev_code=5;
    elseif I(start1,start2-1)==0 %code=4 - prev_code=3
        start1=start1;

```

```

    start2=start2-1;
    store_code{increment}=4;
    prev_code=4;
elseif I(start1-1,start2-1)==0 %code=3 - prev_code=3
    start1=start1-1;
    start2=start2-1;
    store_code{increment}=3;
    prev_code=3;
elseif I(start1-1,start2)==0 %code=2 - prev_code=3
    start1=start1-1;
    start2=start2;
    store_code{increment}=2;
    prev_code=2;
elseif I(start1-1,start2+1)==0 %code=1 - prev_code=3
    start1=start1-1;
    start2=start2+1;
    store_code{increment}=1;
    prev_code=1;
elseif I(start1,start2+1)==0 %code=0 - prev_code=3
    start1=start1;
    start2=start2+1;
    store_code{increment}=0;
    prev_code=0;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==4 % ***** if prev_code==4
    if I(start1+1,start2-1)==0 %code=5 - prev_code=4
        start1=start1+1;

```



```

    start2=start2-1;
    store_code{increment}=5;
    prev_code=5;
elseif I(start1,start2-1)==0 %code=4 - prev_code=4
    start1=start1;
    start2=start2-1;
    store_code{increment}=4;
    prev_code=4;
elseif I(start1-1,start2-1)==0 %code=3 - prev_code=4
    start1=start1-1;
    start2=start2-1;
    store_code{increment}=3;
    prev_code=3;
elseif I(start1-1,start2)==0 %code=2 - prev_code=4
    start1=start1-1;
    start2=start2;
    store_code{increment}=2;
    prev_code=2;
elseif I(start1-1,start2+1)==0 %code=1 - prev_code=4
    start1=start1-1;
    start2=start2+1;
    store_code{increment}=1;
    prev_code=1;
else
    start1=0;
    start2=0;
end
increment=increment+1;
elseif prev_code==7 % ***** if prev_code==7
    if I(start1-1,start2+1)==0 %code=1 - prev_code=7
        start1=start1-1;

```

```
start2=start2+1;
store_code{increment}=1;
prev_code=1;
elseif I(start1,start2+1)==0 %code=0 - prev_code=7
start1=start1;
start2=start2+1;
store_code{increment}=0;
prev_code=0;
elseif I(start1+1,start2+1)==0 %kod=7 - prev_code=7
start1=start1+1;
start2=start2+1;
store_code{increment}=7;
prev_code=7;
elseif I(start1+1,start2)==0 %code=6 - prev_code=7
start1=start1+1;
start2=start2;
store_code{increment}=6;
prev_code=6;
elseif I(start1+1,start2-1)==0 %code=5 - prev_code=7
start1=start1+1;
start2=start2-1;
store_code{increment}=5;
prev_code=5;
elseif I(start1,start2-1)==0 %code=4 - prev_code=7
start1=start1;
start2=start2-1;
store_code{increment}=4;
prev_code=4;
else
start1=0;
start2=0;
```

```

end
increment=increment+1;
end %end for prev_code=5

end % end for prev_code=8

end % end for while loop

S = cell2mat(store_code);

disp('Kod rantaian yang dibaca adalah:')
disp(S)

```

5.4.2 Run length coding

Setelah mendapatkan kod rantaian bagi imej, algoritma *run-length coding* dilaksanakan bagi mengurangkan *redundancy* pada bacaan kod rantaian. Algoritma adalah seperti berikut:

```

[o,p]=size(S); %mengambil parameter S daripada bacaan kod rantaian
Z=cell(o,1);
tempi=0;
tempii=1;
i=0; %membezakan turutan kod rantaian yang dibaca
for j=1:p-1
    tempi=j; %tempi memegang nilai j
    a=S(1,tempi);
    b=S(1,tempi+1);
    if a<=b
        T=b-a;
    else

```



```

    T=a-b;
end
if (T==0)
    Z{tempii}=a;
else
    tempii=tempii+1;
    Z{tempii}=b;
end
tempi=j;
end
disp('Run length coding ialah: ')
z=cell2mat(Z);
disp(z)

```

5.4.3 Pengecaman Bit

Daripada proses pensegmenan:

```
z=cell2mat(aksara_tambahan);
```

Oleh itu, algoritma pengecaman bit adalah seperti berikut:

```

[m,n]=size(z);
for i=1:m
    for j=1:n
        if z(i,j)==0
            temp1=i;
            temp2=j;
            break;
        end
    end
end
temp3=temp1;
temp4=temp2+1;

```

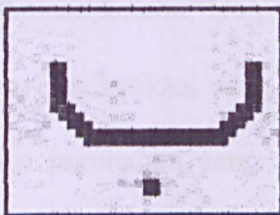
```
for j=temp4:n
    if z(temp3,j)==1
        figure,imshow(z),title('Satu Titik');
        temp5=j;
        break;
    end
end
```

```
temp6=temp5+1;
```

```
for j=temp6:n
    if z(temp3,j)==0
        figure,imshow(z),title('Dua Titik')
    end
end
```


6.1 PENGUJIAN SISTEM

Dalam melaksanakan pengujian ke atas alatan yang telah dibangunkan, algoritma-algoritma yang dibangunkan akan diuji dan diimplementasikan ke atas imej huruf jawi tunggal. Output imej bagi setiap tahap pengujian akan dihantar sebagai input ke tahap seterusnya. Bagi melakukan pengujian, imej huruf jawi 'ba' telah digunakan seperti dalam *Rajah 11*.



Rajah 11: Imej huruf 'ba' yang akan dijadikan bahan ujian.

Ujian dilakukan berdasarkan modul-modul berikut:

6.2 PEMROSESAN IMEJ

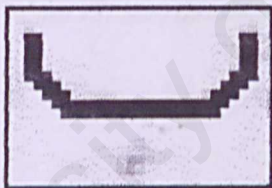
6.2.1 Pendigitalan Imej

Fungsi `[imagefile,imagepath]=uigetfile('*. *','Select your image')` akan memulangkan nama dan laluan bagi fail yang dipilih di dalam kotak direktori. Selepas butang *Open* ditekan, *imagefile* mengandungi nama bagi fail yang telah dipilih dan *imagepath* mengandungi nama laluan yang dipilih. Jika butang *Cancel* ditekan atau jika ralat terjadi, *imagefile* dan *imagepath* diset kepada 0. Kemudian, fungsi `[I,map]=imread([imagepath,imagefile])` akan membaca indeks imej pada *imagepath* kepada *I* dan *map* yang berkaitan dengannya kepada *map*. Nilai *map* adalah diskalakan kepada julat [0,1]. *imagepath* adalah string yang menspesifikkan format bagi fail. Jika *imread* tidak boleh mencari nama fail *imagepath*, ia akan lihat nama fail *imagepath.fmt*.

6.2.2 Pengurangan Gangguan

Fungsi $B = \text{rgb2gray}(I)$ digunakan untuk menukarkan nilai sebenar bagi imej I kepada intensiti tahap kelabu bagi imej B . Di dalam pengurangan gangguan imej, fungsi medfilt2 adalah untuk melaksanakan dwi-dimensi penapis median. Fungsi $\text{noise} = \text{medfilt2}(B, [3\ 3])$ menjalankan proses penapisan median bagi dwi-dimensi matriks B , di mana B mewakili imej penskalaan kelabu. Output bagi setiap piksel mengandungi nilai median dalam piksel kejiranan 3×3 piksel yang diinput. Piksel berkejiranan 3×3 digunakan kerana ia tidak memberi perubahan yang terlalu ketara sehingga mengubah imej asal. Ia mampu mengurangkan gangguan yang berlaku pada imej.

Imej huruf jawi yang diinput dari fail akan melalui fungsi ini untuk mengurangkan gangguan imej dan menghasilkan imej seperti Rajah 12.



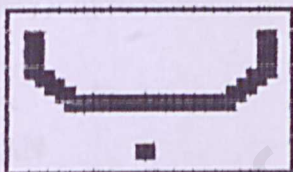
Rajah 12: Imej adalah hasil dari fungsi penapis median

6.2.3 Peningkatan Kualiti Imej

Peningkatan kualiti imej adalah menggunakan kaedah *brightness adjustment*. Fungsi $\text{brighten}(\text{beta})$ digunakan untuk menggantikan *colormap* semasa dengan *colormap* yang cerah atau gelap bagi warna sama. Fungsi $\text{brighten}(\text{beta})$, digunakan bersama fungsi $\text{brighten}(-\text{beta})$, di mana beta adalah kurang daripada 1 dan akan menyimpan *map* asal.

6.2.4 Threshold

Seperti yang diketahui, threshold diperlukan untuk menukarkan imej penskalaan kelabu kepada imej binari. Oleh itu, fungsi *graythresh* digunakan memandangkan ia adalah satu fungsi yang mengira imej *threshold* dengan menukarkan imej tahap kelabu ke imej binari bersama fungsi *im2bw*. Fungsi *threshold = im2bw(noise,level)* akan menukarkan imej intensiti *noise* kepada warna hitam dan putih. Umpukan *level=graythresh(I)*; dilakukan di mana *level* adalah nilai intensiti dalam julat [0 1]. Imej dari output peningkatan kualiti imej digunakan bagi mengeluarkan output seperti di bawah.



Rajah 13: Imej threshold

6.2.5 Thinning

Konsep *dilation* dapat diuji dengan menggunakan beberapa fungsi yang sedia ada di dalam *dilation* itu sendiri seperti *line*, *square*, *diamond* dan lain-lain. *Dilation* adalah operasi di mana nilai bagi sesuatu output piksel adalah nilai maksimum bagi keseluruhan input piksel dalam piksel kejirannya. Fungsi *SE = strel('square',2)* akan dilaksanakan berdasarkan parameter dan bentuk. Di dalam pengujian ini, bentuk *square* digunakan untuk membentuk penstrukturan elemen *square* di mana lebar bagi bentuk *square* ditentukan oleh parameter yang bernilai 2. Fungsi *thinning = imdilate(threshold,SE)* akan menipiskan imej tahap kelabu, imej binari, atau imej *packet binary* bagi imej *SE*, dan akan memulangkan imej nipis *thinning*. Parameter *SE* adalah

objek bagi penstrukturan elemen, atau tatasusunan bagi objek penstrukturan elemen yang dipulangkan oleh fungsi *strel*.

Oleh kerana input bagi modul ini adalah dari modul *threshold* di mana nilai piksel terdiri daripada 0 dan 1, maka jika sesuatu piksel bernilai 1, output bagi piksel tersebut akan diset kepada 1. Hasil dari fungsi ini ke atas input imej *threshold*.



Rajah 14: Imej yang telah melalui fungsi *thinning*

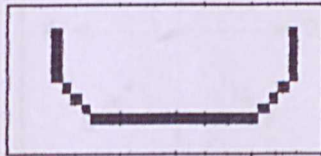
6.3 PROSES PENSEGMEANAN

Proses pensegmenan melibatkan pemecahan aksara asas dan aksara tambahan. Pengujian proses pensegmenan melibatkan satu huruf jawi iaitu *ba* (ب). Imej huruf *ba* akan diinput ke dalam modul pensegmenan hasil dari output modul *thinning*. Proses pensegmenan bermula apabila nilai bagi baris pertama mempunyai sekurang-kurangnya satu piksel yang bernilai sifar dipulangkan. Apabila nilai baris tersebut telah dikenalpasti, semua nilai piksel yang menduduki baris itu dan diikuti baris berikutnya akan disimpan ke dalam satu *cell array* sehingga terdapat satu baris pertama yang mempunyai semua nilai piksel bersamaan dengan satu. Imej yang disimpan ke dalam *cell array* ini adalah mewakili struktur aksara asas.

Baris yang mempunyai semua nilai piksel bersamaan dengan satu menunjukkan

terdapat ruang kosong di antaranya. Maka apabila nilai baris tersebut dipulangkan, piksel-piksel bagi baris tersebut dan baris seterusnya akan disimpan pada *cell array* yang lain bagi mewakili struktur aksara tambahan.

Proses pensегmenan berjaya dilaksanakan kerana wujud dua *cell array* yang berlainan yang menyimpan struktur aksara asas dan aksara tambahan. Seterusnya imej yang disimpan dalam dua *cell array* yang berlainan akan melalui proses yang seterusnya iaitu proses pengecaman aksara. Modul ini akan menghasilkan dua imej sebagai output iaitu imej aksara asas dan imej aksara tambahan seperti dalam *Rajah 15* dan *Rajah 16*.



Rajah 15: Aksara asas

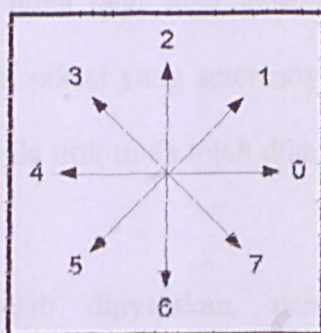


Rajah 16: Aksara Tambahan

6.4 PROSES PENGECAMAN

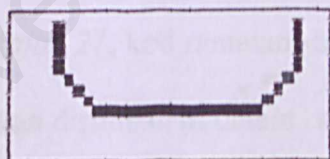
6.4.1 Pembacaan kod rantai

Seterusnya, algoritma pengecaman telah dibangunkan berdasarkan kepada konsep kod rantai. Kod rantai bagi sesuatu imej boleh didefinisikan sebagai titik permulaan imej dan laluan yang dilalui oleh imej tersebut. Laluan bagi imej boleh digambarkan sebagai jujukan arah. Dalam kajian ini, gambarajah arah dengan 8-piksel kejuranan diimplementasikan di dalam mendapatkan kod rantai.



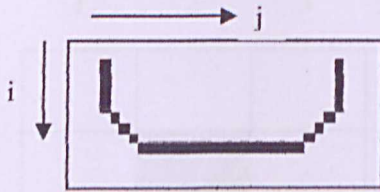
Rajah 17: Gambarajah lapan arah kod rantai

Ujian kod rantai hanya dilakukan ke atas aksara asas bagi huruf 'ba' input daripada modul pensegmenan seperti *Rajah 18*.

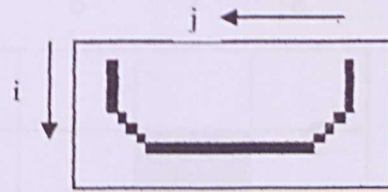


Rajah 18: Input aksara asas dari modul pensegmenan

Sebelum membaca kod rantai, titik mula bagi imej dalam *Rajah 18* perlu ditentukan terlebih dahulu. Imej aksara Jawi boleh dibaca dalam matriks (i, j) di mana i mewakili baris dan j mewakili lajur bagi imej. Penentuan titik mula aksara asas mestilah dibaca dari arah kanan ke kiri berdasarkan tulisan Jawi ditulis dan dibaca.



Rajah 19: Cara pembacaan sebenar baris dan lajur.



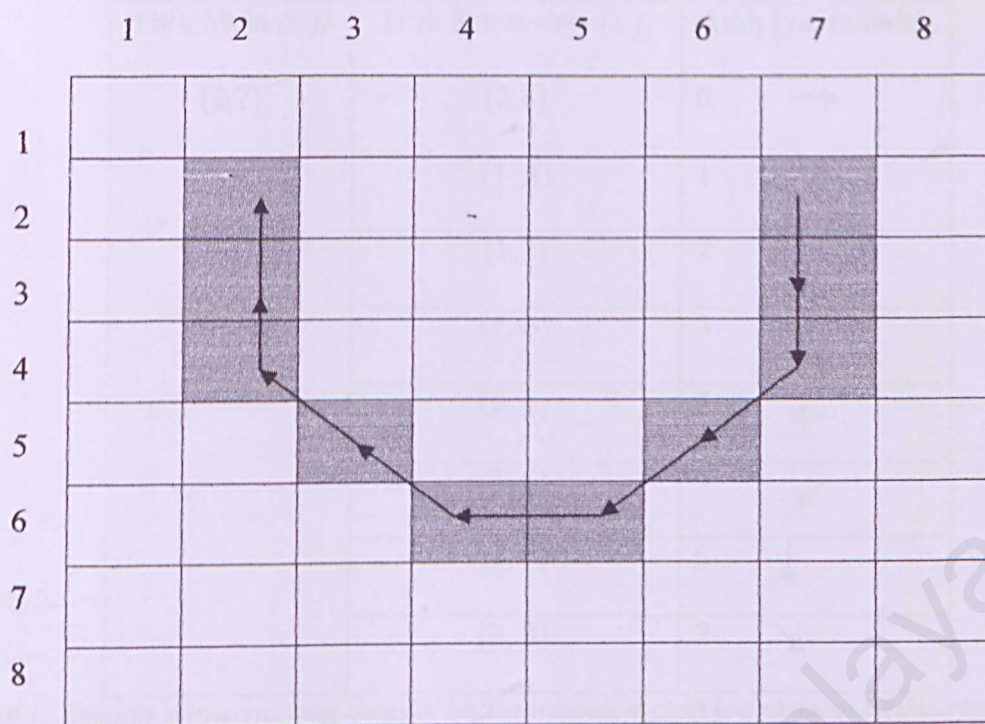
Rajah 20: Cara pembacaan baris dan lajur bagi penentuan titik mula.

Dalam algoritma titik mula, nilai pertama yang dijumpai dalam baris dan lajur (i, j) akan memulangkan nilai titik mula bagi imej aksara asas tersebut. Algoritma akan berhenti daripada membaca nilai piksel yang seterusnya sebaik sahaja menjumpai nilai piksel pertama. Seterusnya, apabila titik mula telah dikenalpasti, pembacaan kod rantaian boleh dimulakan.

Seperti yang telah dinyatakan, pembacaan kod rantaian adalah berdasarkan kepada titik mula dan laluan titik seterusnya. Algoritma kod rantaian akan membaca titik mula dan memeriksa semua laluan yang mungkin yang mempunyai nilai pada piksel 8-kejiranan. Nilai titik mula tersebut akan berubah setiap kali laluan dijumpai. Pengujian di lakukan merujuk kepada *Rajah 21* di bawah.

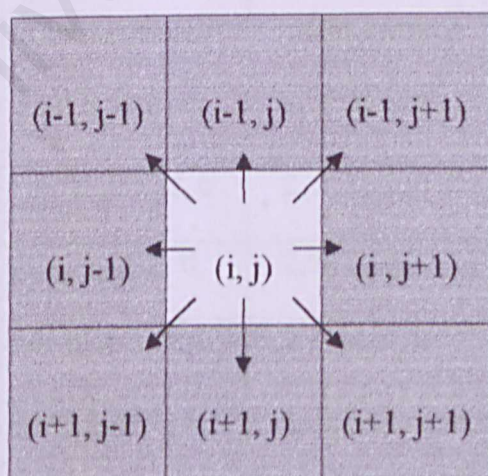
Merujuk kepada *Rajah 21*, kod rantaian yang dibaca ialah 6, 6, 5, 5, 4, 3, 3,

2. Kod rantaian ini kemudian akan disimpan di dalam *cell array*.



Rajah 21: Aksara asas huruf 'ba' di mana piksel berwarna kelabu mewakili nilai piksel 0 (imej) manakala piksel berwarna putih mewakili nilai 1. Anak panah merupakan bacaan kod rantaian

Dalam Rajah 21 di atas, titik mula adalah pada nilai piksel (2,7). Seterusnya berdasarkan kepada titik mula ini, algoritma akan mencari mana-mana piksel kejirannya (8-kejiranan) yang mempunyai nilai 0 (mempunyai imej). Ini digambarkan seperti Rajah 22.



Rajah 22: Nilai piksel-piksel yang mungkin bagi piksel 8-kejiranan

Titik Mula (<i>i, j</i>)	Titik Seterusnya (<i>i, j</i>)	Arah kod rantaian
(2,7)	(2,8)	0 →
	(1, 8)	1 ↗
	(1,7)	2 ↑
	(1, 6)	3 ↖
	(2, 6)	4 ←
	(3, 6)	5 ↙
	(3, 7)	6 ↓
	(3, 8)	7 ↘

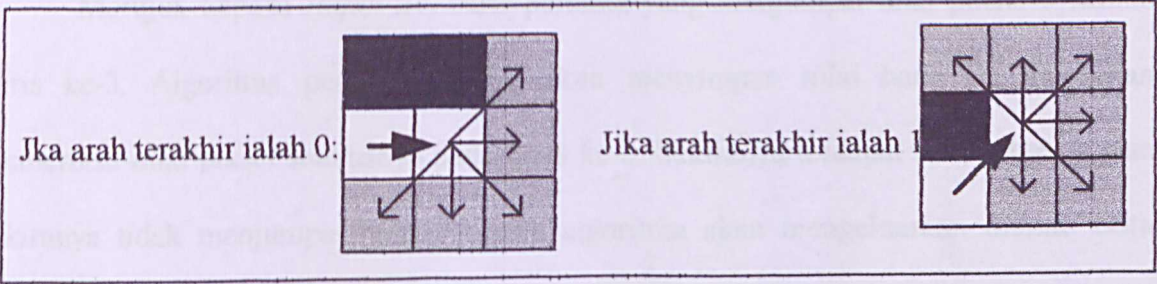
Jadual 1: Titik-titik seterusnya yang mungkin bagi mendapatkan arah kod rantaian berdasarkan titik mula

Semasa pembacaan kod rantaian, adalah ditekankan di sini bahawa tidak semua arah kod rantaian boleh dibaca berdasarkan kepada titik mula. Ini adalah kerana terdapat beberapa kod yang tidak mungkin dilalui oleh kod-kod rantaian. Ini dapat dijelaskan seperti di dalam *Jadual 2* di bawah.

Arah terakhir dibaca	Arah seterusnya (nilai ganjil)	Arah seterusnya (nilai genap)	Arah seterusnya (ikut arah jam)
0	1, 7, 0, 5, 6	0, 2, 1, 6, 7, 5	1, 0, 7, 6, 5
1	3, 1, 2, 7, 0, 6	2, 3, 0, 1, 6, 7	3, 2, 1, 0, 7, 6
2	3, 1, 2, 7, 0	2, 4, 3, 0, 1, 7	3, 2, 1, 0, 7
3	5, 3, 4, 1, 2, 0	4, 5, 2, 3, 0, 1	5, 4, 3, 2, 1, 0
4	5, 3, 4, 1, 2	4, 6, 5, 2, 3, 1	5, 4, 3, 2, 1
5	7, 5, 6, 3, 4, 2	6, 7, 4, 5, 2, 3	7, 6, 5, 4, 3, 2

6	7, 5, 6, 3, 4	6, 0, 7, 4, 5, 3	7, 6, 5, 4, 3
7	1, 7, 0, 5, 6, 4	0, 1, 6, 7, 4, 5	1, 0, 7, 6, 5, 4

Jadual 2: Arah seterusnya yang mungkin berdasarkan kepada arah terakhir dibaca



Rajah 23: Piksel yang berwarna hitam adalah piksel yang tidak mungkin mengandungi imej. Arah anak panah menunjukkan arah-arrah seterusnya yang mungkin mengikut bacaan arah jam merujuk kepada

Jadual 2

6.4.2 Pembacaan run-length coding

Selepas kod rangkaian bagi imej telah dibaca, *cell array* yang menyimpan kod rangkaian tersebut akan ditukar ke dalam bentuk *array*. *Run length coding* adalah bertujuan untuk memadatkan atau meringkaskan pembacaan kod rangkaian untuk mengelakkan pengulangan data atau *redundancy*. Kod rangkaian yang dibaca adalah 6, 6, 5, 5, 4, 3, 3, 2 akan dipadatkan dan diringkaskan kepada 6, 5, 4, 3, 2.

6.4.3 Pengecaman Bit

Proses terakhir dalam modul pengecaman adalah pengecaman bit. Proses ini adalah untuk menilai jumlah titik yang wujud di dalam imej aksara tambahan dari output imej pada modul pensegmenan. Proses ini akan memeriksa dan menyimpan baris bagi nilai piksel pertama yang bernilai 0 yang melambangkan imej. Piksel pertama yang

bernilai 0 melambangkan satu titik. Kemudian, proses ini akan memeriksa nilai seterusnya pada baris tersebut. Sekiranya terdapat nilai piksel 1 dan kemudian nilai piksel 0, maka imej akan dikenali sebagai dua titik.

Merujuk kepada *Rajah 24*, baris pertama yang menjumpai nilai piksel 0 adalah baris ke-3. Algoritma pengecaman bit akan menyimpan nilai baris ini dan akan memeriksa nilai piksel seterusnya pada baris ke-3. Sekiranya terdapat nilai piksel 1, dan sekiranya tidak menjumpai nilai 0, maka algoritma akan mengeluarkan output 'Satu titik'. Namun dalam contoh *Rajah 24* di bawah, terdapat nilai piksel 0 selepas itu, maka output akan mengeluarkan 'Dua titik'.

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	0	0	1	0	0	1	1
4	1	0	0	1	0	0	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1

Rajah 24: Warna kelabu melambangkan imej aksara tambahan yang mewakili dua titik. Nilai 0 dan 1 di dalam piksel melambangkan nilai yang dipegang oleh setiap piksel di mana 0 mewakili warna hitam (imej) dan 1 mewakili warna putih.

7.1 PERBINCANGAN

Kepentingan yang dipentaskan daripada aliran perancangan jawi yang dibangunkan ialah perancangan perancangan yang menggunakan idea, dan perancangan ini.

Sepanjang membangunkan aliran perancangan jawi ini, beberapa kelebihan dan kelemahan yang dapat dijumpai oleh perancang dengan hasil projek ini.

Perancang mengkaji kepada perancang yang lain yang telah dibangunkan di dalam aliran ini. Cara

BAB 7

PERBINCANGAN

Perancang akan mengkaji kepada perancang yang lain yang telah dibangunkan di dalam aliran ini. Cara

Perancangan dalam perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini. Perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini.

Perancangan dalam perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini. Perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini.

Perancangan dalam perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini. Perancangan jawi ini adalah dengan perancangan yang menggunakan idea, dan perancangan ini.

7.1 PERBINCANGAN

Keputusan yang diperolehi daripada alatan pengecaman jawi yang dibangunkan ialah perlaksanaan pemprosesan imej, pensегmenan imej, dan pengecaman imej.

Sepanjang membangunkan alatan pengecaman jawi ini, beberapa kelebihan dan kelemahan telah dapat dikenalpasti jika dibandingkan dengan hasil projek sebelum ini. Terdapat kelebihan kepada penambahan ciri-ciri dan idea baru diimplementasikan di dalam alatan ini. Ciri-ciri baru tersebut disenaraikan seperti berikut:

- Membina alatan pengecaman imej yang berfungsi dalam pemprosesan imej termasuk pengurangan gangguan imej, peningkatan kualiti imej, proses *thresholding*, dan proses *thinning*.
- Pelaksanaan alatan pengecaman jawi berfungsi dalam proses pensегmenan tanpa mengambil kira kedudukan aksara jawi sama ada ia terletak di bahagian atas, tengah, atau bawah. Pensегmenan yang baru dibina berjaya memecahkan aksara jawi kepada aksara asas dan aksara tambahan secara automatik, dan bukan secara manual.
- Pelaksanaan alatan pengecaman jawi mampu berfungsi dalam proses pengecaman yang melibatkan pengecaman aksara asas dan aksara tambahan. Penemuan titik mula yang diperolehi adalah amat penting dalam mengimplementasikan pembacaan kod rangkaian. Algoritma kod rangkaian yang dibina dalam projek ini dapat dilaksanakan dengan berjaya dan ia merupakan algoritma yang paling penting dalam menentukan perlaksanaan alatan ini dalam mengecam aksara jawi.
- Membina alatan untuk proses *run-length coding* yang belum pernah diimplementasikan oleh projek sebelum ini memberi faedah yang besar dalam

alatan ini. *Run-length coding* memberi faedah dalam menyimpan kod rangkaian yang dibaca tanpa pengulangan data.

- Pembangunan alatan pengecaman jawi untuk proses pengecaman aksara tambahan yang tidak mempengaruhi faktor kedudukan koordinat seperti dilaksanakan oleh projek sebelum ini. Pengecaman aksara tambahan dalam projek ini mampu mengecam aksara tambahan dengan pembinaan algoritma pintar.

Kelemahan utama yang terdapat dalam alatan pengecaman jawi ini ialah tiada penghantaran parameter secara keseluruhan dalam modul pemprosesan imej. Ini membawa kepada ketidakebekersanan dalam proses *thinning*.

Selain daripada itu, fungsi untuk pengurangan gangguan serta pengubahsuaian tahap kecerahan imej yang dibina dalam modul pemprosesan imej adalah bertujuan untuk pembuktian fungsi tersebut berjaya tanpa dihantar kepada proses seterusnya.

Dalam proses pengecaman aksara asas, imej yang diinput mestilah diwakili dengan satu piksel mewakili satu garis lurus. Kod rangkaian tidak dapat dibaca sekiranya syarat ini tidak dipenuhi.

Tahap pengujian yang dilakukan adalah menggunakan satu huruf aksara jawi tunggal *ba* ب. Ini tidak menepati skop yang telah ditetapkan bertujuan untuk menguji pelaksanaan algoritma dan membina algoritma. Selain dari itu, rekabentuk antaramuka pengguna yang telah dicadangkan berlainan dengan antaramuka pengguna yang dibina kini. Ini adalah kerana untuk memudahkan pengguna melihat proses yang berlaku dengan lebih jelas.

Peningkatan yang boleh dilakukan pada masa hadapan ialah dengan memastikan proses *thinning* dapat dijalankan dengan menggunakan input imej dari proses yang sebelumnya dengan menghantar parameter. Selain dari itu, dalam pembacaan kod rantaian dalam modul pengecaman, satu algoritma *fault tolerance* dapat dibina dengan memilih dan membaca kod rantaian sebenar sungguhpun pembacaan kod rantaian adalah salah sekiranya imej yang diinput tidak menepati syarat satu piksel mewakili satu garis lurus.

Saya mencadangkan agar projek ini dapat diteruskan dengan mengatasi kelemahan seperti yang dinyatakan di atas dan menambah ciri-ciri baru bagi membina alatan pengecaman aksara jawi dengan lebih efisien supaya perisian alatan ini boleh dikombinasikan bersama perkakasan pengecaman aksara jawi.

Kesimpulannya, terdapat peningkatan fungsi dalam alatan pengecaman jawi pada projek ini berbanding projek sebelumnya dan penghasilan algoritma-algoritma yang amat penting dalam menjayakan alatan ini

RUJUKAN

Adnan Amin. (2002). Structural Description to Recognising Arabic Characters Using Decision Tree Learning Techniques. *Springer-Verlag Berlin Heidelberg*. **2396**.152-158.

Adnan Amin. (2001). Segmentation of Printed Arabic Text. *Springer-Verlag Berlin Heidelberg*. **2013**.115-126.

Adnan Amin. (1998). Off-Line Arabic Character Recognition: The State of the Art. *Pattern Recognition*. **31(5)**. 517-530.

M. S. Khorsheed. (2002). Off-Line Arabic Character Recognition –A Review. *Pattern Analysis & Applications*. **5**.31-45

Omar Abdul Rahim. (2001). *Jawi Character Recognition*. Bsc. Thesis. University of Malaya.

Scott E Umbaugh. (1998). *Computer Vision and Image Processing*. Prentice Hall.

<http://www.mathworks.com/access/helpdesk/help/toolbox/images/images.shtml>

http://www.dai.ed.ac.uk/HIPR2/hipr_top.htm

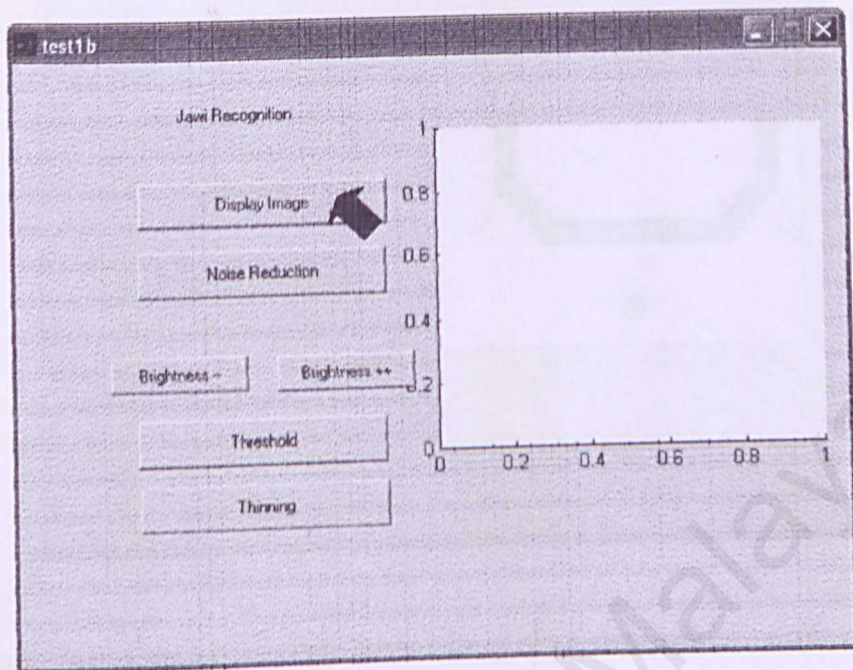
<http://www.dai.ed.ac.uk/CVonline/transf.htm>

LAMPIRAN

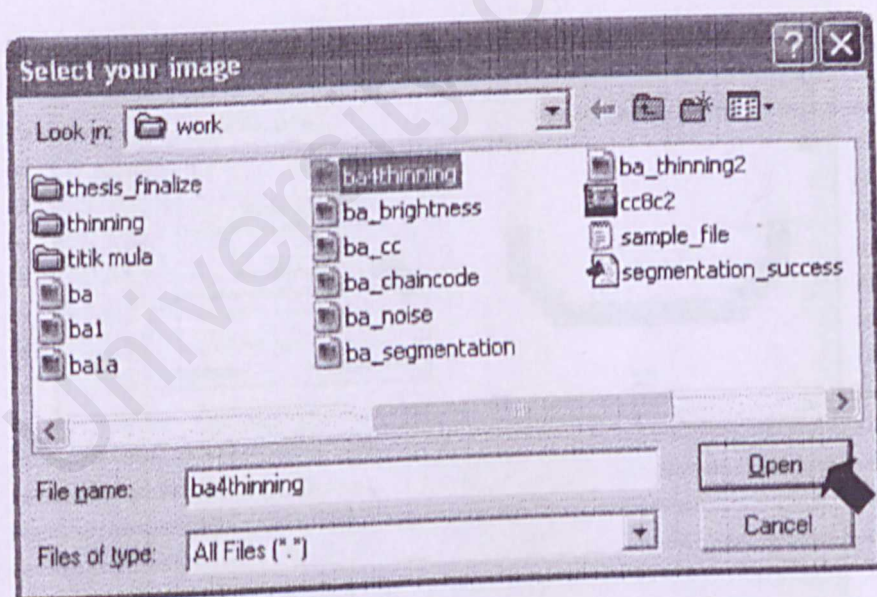
University of Malaya

MODUL PEMROSESAN IMEJ

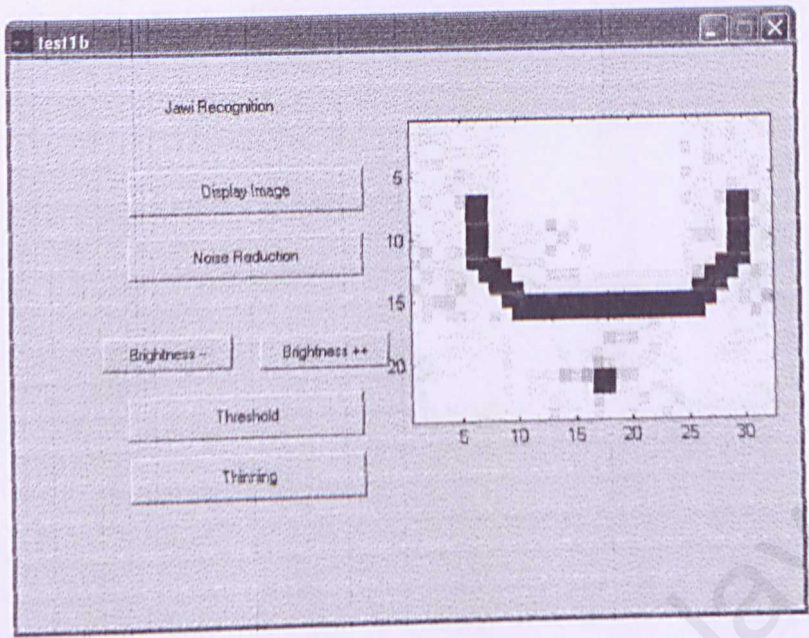
1. Klik butang *Display Image* dan *pop-up window* akan dipaparkan.



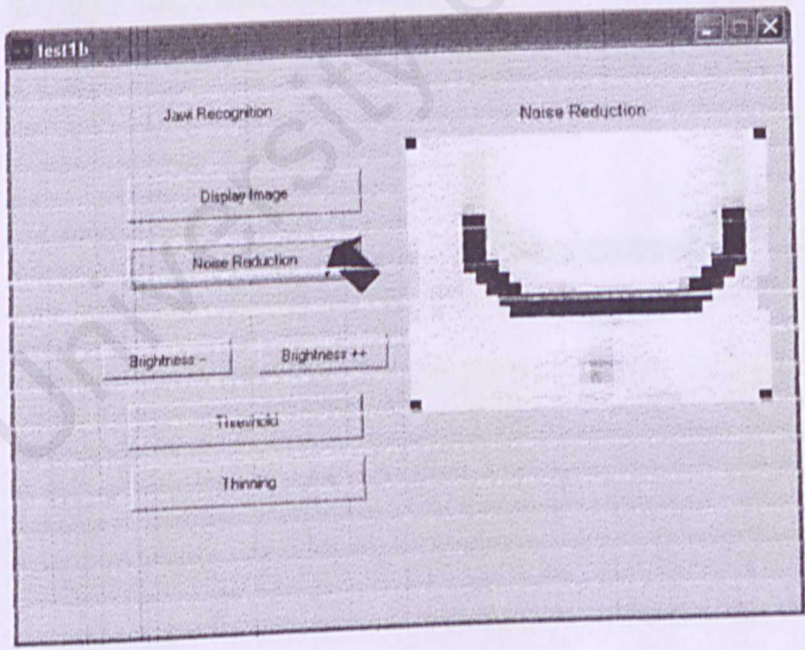
2. Pilih fail imej dari direktori dan klik *Open*.



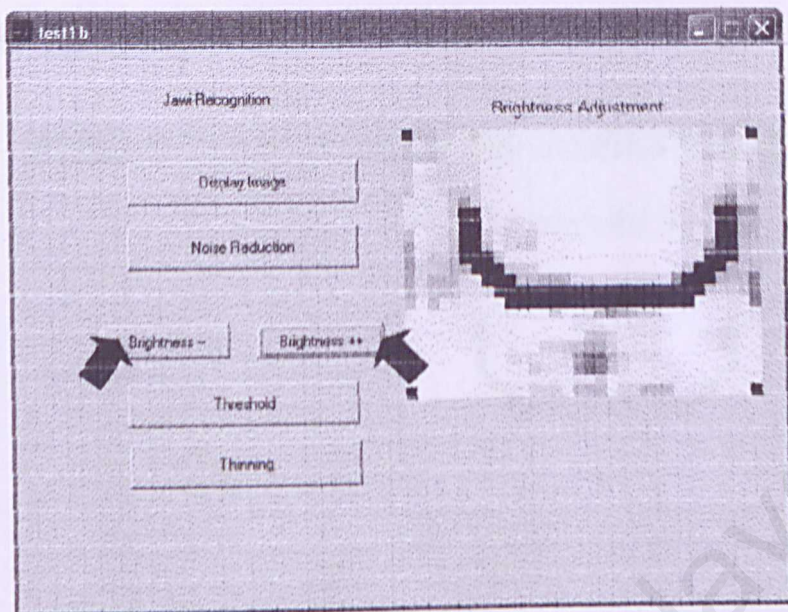
3. Imej bagi imej fail yang dipilih akan dipaparkan.



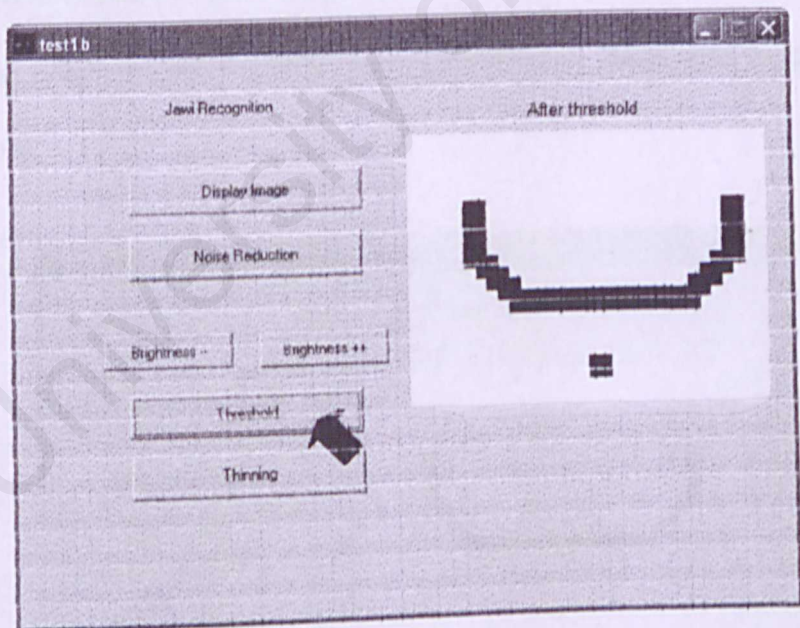
4. Klik butang *Noise Reduction* dan output imej akan dipaparkan untuk mengurangkan gangguan imej menggunakan penapis *median*.



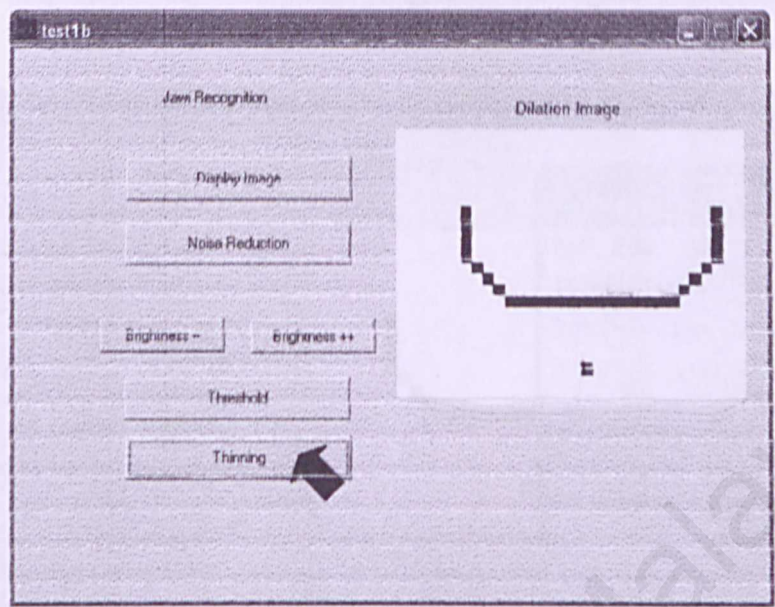
5. Klik butang *Brightness --* dan *Brightness ++* untuk mengubah tahap kecerahan imej.



6. Butang *Threshold* diklik bagi melaksanakan proses *threshold* yang akan menukarkan imej tahap kelabu ke imej binari dan output akan dipaparkan.

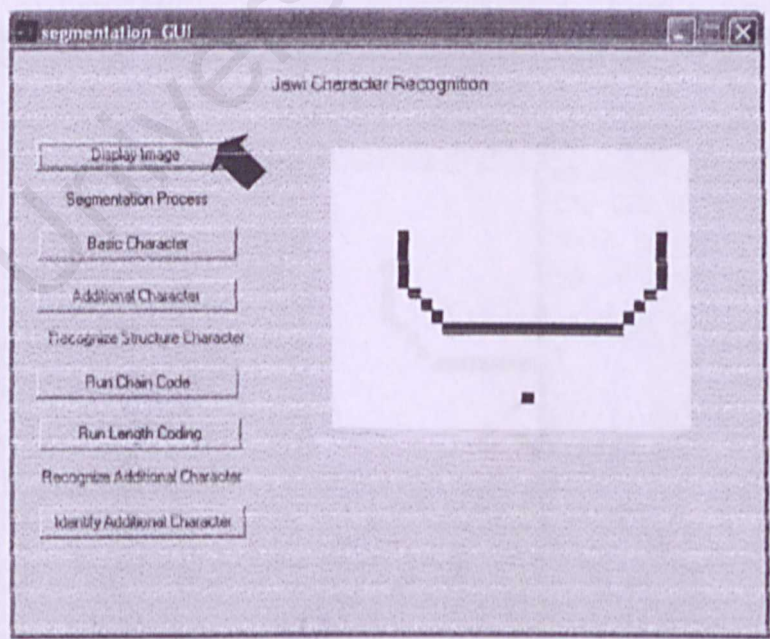


7. Klik butang *Thinning* untuk memaparkan output selepas proses *thinning* dilaksanakan.
- Proses *thinning* dilaksanakan untuk menghasilkan imej garis lurus.



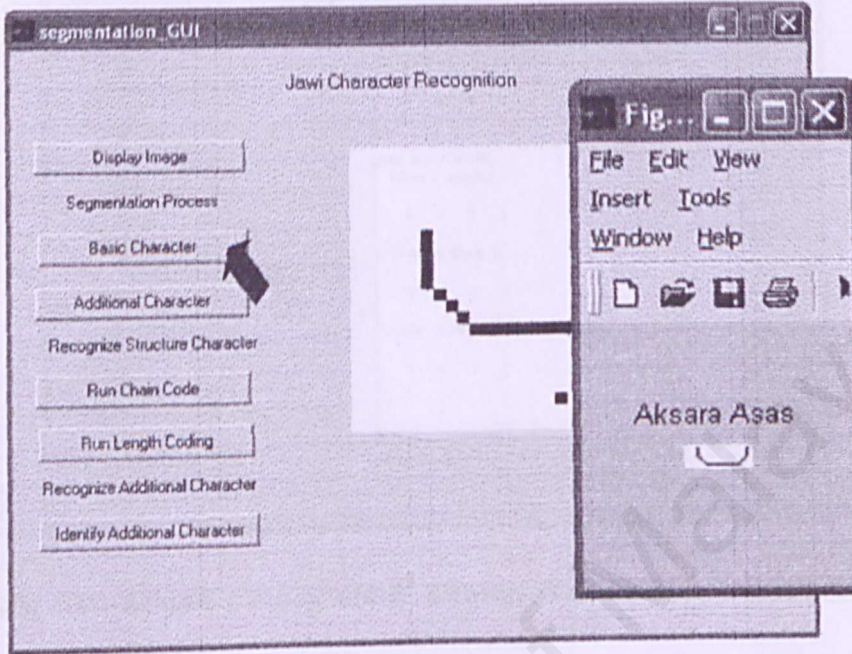
MODUL PENGECAMAN AKSARA JAWI

1. Klik butang *Display Image* untuk memaparkan imej hasil dari output imej dalam proses *threshold*.

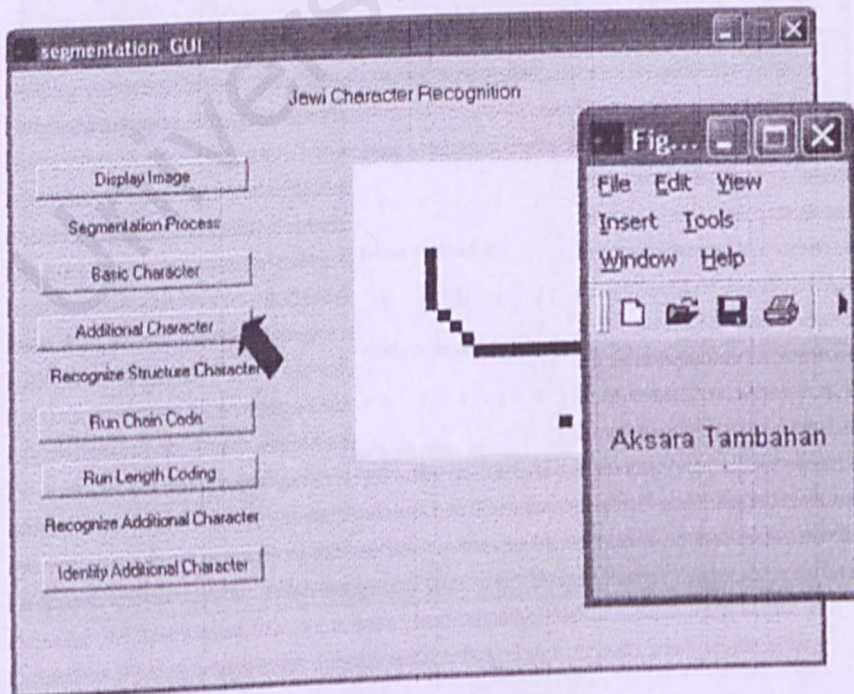


Proses Pensegmenan

2. Untuk meneruskan dengan proses pensegmenan, klik butang *Basic Character* untuk melihat dan memaparkan imej aksara asas yang telah diasingkan dari imej asal.

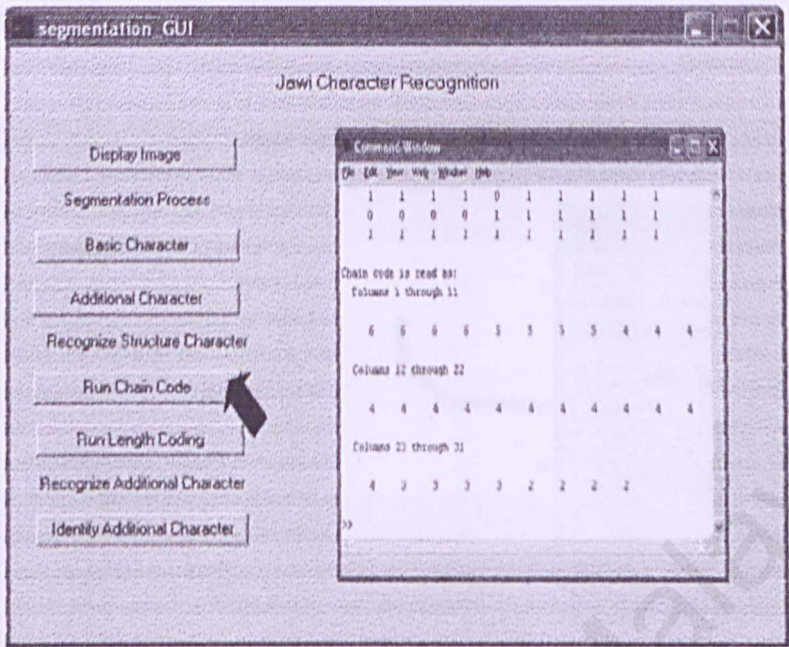


3. Proses yang sama berlaku apabila menekan butang *Additional Character* yang akan memaparkan imej aksara tambahan yang telah tersegmen.

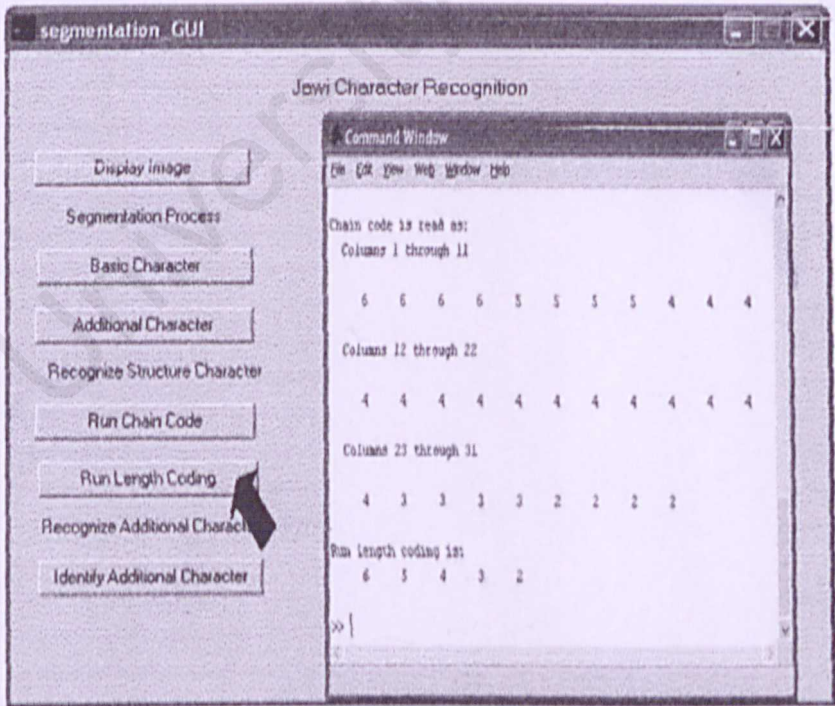


Proses Pengecaman Aksara Asas

1. Klik butang *Run Chain Code* untuk melihat bacaan kod rantai bagi aksara asas.



2. Klik butang *Run Length Coding* untuk memaparkan bacaan kod rantai yang telah diringkas dan dipadatkan.



Proses Pengecaman Aksara Tambahan

1. Klik butang *Identify Additional Character* untuk mengecam bilangan titik yang dipunyai oleh aksara jawi.

